SIMULATION FEASIBILITY STUDIES OF A HIGH DENSITY FNIRS IMAGING SYSTEM

by

PALLAVI NATEKAR

Presented to the Faculty of the Graduate School of

The University of Texas at Arlington in Partial Fulfillment

of the Requirements

for the Degree of

MASTER OF SCIENCE BIOMEDICAL ENGINEERING

THE UNIVERSITY OF TEXAS AT ARLINGTON

August 2014

Acknowledgements

I feel a sincere sense of gratitude towards Dr. George Alexandrakis for offering me a chance to work on his vision and guiding, encouraging and challenging me throughout. He has been extremely understanding and allowed me to self-explore and learn. He has always offered support and resources and I am highly inspired by him and his work.

I am extremely thankful to Dr. Liu, Dr. MacFarlane and Dr. Alexandrakis for being on my thesis committee despite their busy schedules and offering me their invaluable advice.

I want to thank Bilal Khan for mentoring me and being extremely patient with my pace of learning during the entire course of the project. I also want to express my deepest gratitude towards my lab-mates Nathan and Jian-Wei for always being there to address my problems and offer every help they could.

I also want to recognize the contribution of my friends who never failed to cheer me up and were my strongest support system. I feel immensely blessed to have Mayur by my side through thick and thin and kept me going. Finally, I will forever remain indebted towards my parents Surendra Natekar, Shweta Natekar and my sister Ketaki for making the most unconventional choices to enable me realize my ambitions.

June 27, 2014

Abstract

SIMULATION FEASIBILITY STUDIES OF A HIGH DENSITY FNIRS IMAGING SYSTEM

Pallavi Natekar, M.S.

The University of Texas at Arlington, 2014

Supervising Professor: Dr. George Alexandrakis

Functional near infrared spectroscopy (fNIRS) is a technique that enables monitoring brain function, by detecting changes in near-infrared light absorbance resulting from changes in brain hemodynamics during periods of rest and activation. Although the technique has some advantages over other conventional functional imaging methods with its inherently high temporal sampling and detection sensitivity, the sparse spatial sampling of current fNIRS systems limits spatial resolution and requires help from Magnetic Resonance Imaging (MRI) to attain spatial co-registration with anatomical structures. In this work we have performed simulations to test the feasibility of using a high density spatial sampling system as a means to detect anatomical structures on the brain surface using near-infrared light signals only while also significantly improving the efficiency of signal detection.

We chose to perform our simulation using a Monte Carlo eXtreme code, which is an accelerated GPU-based Monte Carlo simulation technique that resulted in computational time gains of 300X over a traditional code on a non-GPU set-up. This acceleration is achieved due to high thread parallelization and improved memory latency that speeds up the algorithm to trace multiple simulated photons in parallel. With the vision to create a simulation set-up that mimics reality as much as possible, we sourced an anatomical brain model from MRI and included hemodynamic fluctuations in different

tissue compartments as explained below. The 3-D image volume was spatially processed in Statistical Parametric Modeling software and subsequently in MATLAB to transform it into a simulated tissue model. The simulation set-up was designed to place a dense 1mm grid of detector fibers on the scalp. Tissue optical properties were defined at the common fNIRS wavelength of 830 nm. An estimate for an adequate simulated photon number to make the stochastic noise from Monte Carlo smaller than the amplitude of hemodynamic fluctuations was made by running independent trials on the tissue model and analyzing the standard error between trials.

A resting state brain model was considered to be appropriate for testing the feasibility of detecting cortical sulci by high density fNIRS since background hemodynamics are known to be present during all times. In order to create the hemodynamic background, information of common sources of physiological hemodynamics, namely Mayer waves (~0.1Hz) and respiratory waves (0.2 – 0.4 Hz) was sourced from the fMRI BOLD data coregistered with the anatomical MRI image volume used previously. The fMRI dataset had lower temporal sampling (0.5 Hz) and hence only Mayer waves and not respiratory waves could be sourced from the fMRI data. Respiratory waves from an fNIRS baseline data set sampled at 10.35 Hz were introduced into the brain tissue model at the scalp and gray matter in a depth-independent manner. The power spectra of hemodynamic fluctuations from Mayer waves and respiration were combined into a consistent single power spectrum and were added to the brain tissue voxels. The resulting hemodynamic fluctuation data were used to modify the light absorbance simulated by Monte Carlo and create resting state time-series reflectance data for the simulated high density fNIRS system. Four simulations for sources at the corners of a 27 mm sized source paired with detectors along a circle of 24 mm radial distance were chosen as the best geometry for reconstructing 2-D cortical hemodynamic

fluctuation maps. We analyzed the resulting images by a signal cross-correlation method and were able to identify cortical sulci from gyri within the center of the imaging field of view. Interestingly this approach could detect sulci that were even 2mm deep form the cortical surface. These preliminary results show that it may be worthy building a very high density fNIRS for mapping anatomical features along with activation maps.

Furthermore, to quantify the benefits of dense spatial sampling on signal collection, a simulated activation region was embedded into the central sulcus of the sensorimotor region in the cortex. Appropriate hemodynamic response functions for this activation region were designed for finger tapping at 1 Hz for 16 sec, 8 sec and 4 sec. Detector fibers in the proposed system were grouped to determine an effective detector diameter size of 13 mm as the most appropriate for maximizing the activation signal-to-noise ratio of activation. Compared with sparser spatial sampling from a conventional fNIRS system, the high density system offered gains of 125% - 400% in signal-to-noise ratio depending on detector placement with respect to the activation location. Also, the dense spatial sampling system showed prospects of reducing the total duration of an activation protocol by half.

Finally, photon budget calculations demonstrated the feasibility of collecting adequate signal from a single detector fiber while staying within light power exposure safety limits, which would have to be taken into account in a real life system. The simulation feasibility studies performed here show that a high density sampling systems holds potential for revolutionizing the fNIRS field.

Table of Contents

List of Illustrations

List of Tables

Chapter 1

Introduction

1.1 Brain Activation

1.1.1 Neurovascular Coupling

The brain works constantly in the resting state to regulate physiological activities in the body as well to coordinate the responses to external stimuli. As a result of neuronal activation an increased flow in the arteriolar bed is seen to spread in the activating areas of the cortex thereby increasing the availability of glucose and oxygen to the activating neurons through those blood vessels {Bunce, 2006 #1234}. The brain requires a steady supply of oxygen during normal processes and a surplus of oxygen to fuel neuronal activity. An overcompensation of oxygenated blood in the local arteriolar bed results with this increased demand for oxygen {Buxton, 2001 #1210} {Ferrari, 2012 #1215}. Hemoglobin is the oxygen transporter protein in blood and oxy hemoglobin (HbO) and deoxy hemoglobin (Hb) are the two forms of hemoglobin when bound with and without oxygen respectively. At times when the requirement for glucose and oxygen increases due to increased activity, the levels of glucose and oxygen drop causing local arteriolar dilation and subsequently increased cerebral blood flow and cerebral blood volume {Ferrari, 2012 #1215}. The underlying phenomenon that causes change in concentration levels of HbO and Hb as a result of neuronal activity is called ‘Neurovascular Coupling’ {Bunce, 2006 #1234}. Thus, an increased neuronal activity results in increased cerebral oxygenation and neurovascular coupling is the principle basis of using optical imaging techniques to study cortical function. An increase in HbO and a decrease in Hb in a spatially localized region is observed in response to a functional task {Strangman, 2002 #1205}.

1.1.2 Sensorimotor Cortex

The integration of the primary motor cortex and primary somatosensory cortex regions the cerebral cortex is commonly termed as the sensorimotor cortex. The central sulcus is the cortical fold between these two regions. The primary motor cortex lies anterior to the central sulcus and is associated with planning and execution of movement in co-ordination with other motor control areas. The somatosensory cortex is positioned posterior to the central sulcus and is the process for receiving and processing senses like touch, temperature, proprioception and pain {Silverthorn, 2012 #1245}. An integration of somatosensory and motor systems exists, called 'Sensorimotor Coupling', because of which a motor response is coupled with a sensory stimulus and vice-versa {Flanders, 2011 #1223}. Senses from the entire body are received by specific regions of the somatosensory cortex and similarly control of body parts for motor action is possible via specific associated regions on the primary motor cortex. Figure 1-3 shows the Cortical Motor Homunculus and Cortical Sensory Homunculus for the sensorimotor cortex.

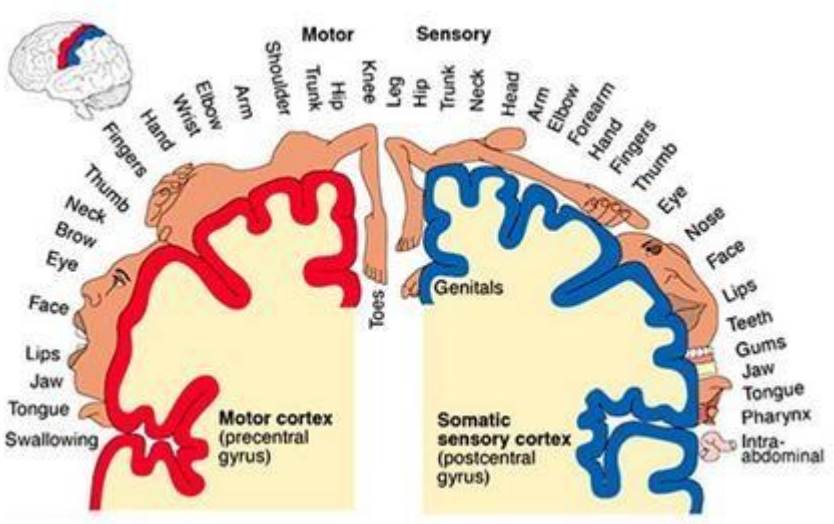

Figure 1-1 Cortical Homunculus - Motor and Somatosensory cortices

[Source: https://neurowiki2012.wikispaces.com/Phantom+Sensations+and+Perceptions]

This work is focused on the somatosensory cortical region. By use of the 10-20 electrode set-up the somatosensory cortex area is defined as being ~ 7.5 cm away from either side of the longitudinal fissure, conventionally marked as C3 and C4, can be identified as the sensorimotor cortex anatomically in the brain {Flanders, 2011 #1223}. This region that is primarily associated with control and sensation from hand digits.

In this work we are studying neuronal activity by simulating changes in the hemodynamics of the primary motor cortex. This hemodynamic activation which is a result of neurovascular coupling is detected by an optical method that is explained here below.

1.2 Principle of Near Infrared Spectroscopy

1.2.1 Functional Near Infrared Spectroscopy

FNIRS is a non-invasive and safe technique that capitalizes on the difference in absorption spectra of HbO and Hb to study brain functioning in vivo using more than one wavelength in the Near Infrared (NIR) 'Optical Window' where their spectra differ {A P Gibson1, 2005 #1247}. A typical fNIRS set-up consists of a fiber optic source and detector optodes placed in contact with the subject's scalp along with the associated electronics. The photons from the illumination are transmitted along the various layers of the tissue and interact with the HbO and Hb primarily at NIR wavelengths on their trajectory before finding their way to the fiber optic detectors positioned to collect the transmitted light. Interaction of photons with these chromophores is important because studies have indicated their association with neuronal activity by the phenomenon of 'Neurovascular Coupling' {Bunce, 2006 #1234} discussed in section 1.1.1. The photons reflected at the detectors are the key to study the activity in the region of the brain it traversed. The length of path of the NIR light is longer than the source-detector separation and estimated to be about one half of the source-detector separation. For an

adequate depth of 1.5 cm, measured from scalp, by NIR light to detect activation in the cerebral sub-cortex, the source-detector separation is computed to be 2.5 to 4 cm {Bunce, 2006 #1234} {A P Gibson1, 2005 #1247} {Ferrari, 2012 #1215}. The head regions visited by photon trajectories for shorter spaced detectors and farther spaced ones are pictured in Figure 1-2. FNIRS assumes that scattering co-efficient remains constant while absorption co-efficient varies with changes in concentrations of HbO and Hb in the NIR wavelength region {Montcel, 2005 #1209}.

1.2.2 Baseline Hemodynamic Fluctuations

During resting periods, fluctuations in concentration of HbO and Hb are present in tissue types that are perfused with blood supply, namely the gray matter and scalp regions. These fluctuations occur irrespective of any cortical activation due to presence of hemodynamic fluctuations along the blood vessels. Commonly observed physiological sources of resting state hemodynamics are cardiac and blood pressure pulsations, breathing and arterial pulsatile pressure {Tong, 2011 #1236} {Tong, 2010 #1222}. The Mayer waves are arterial pressure waves that are coupled with sympathetic nervous activity {Julien, 2006 #1219} and categorized at low frequency oscillations (LFO) in the range of 0.07 – 0.2 Hz) {Vermeij, 2014 #1211}. These waves have a stable frequency of ~0.1 Hz in humans irrespective of cortical activity. Spontaneous Very Low frequency Oscillations (VLFO) in the range of 0.02 – 0.07 Hz are also known to be present. These waves get stronger with cortical activation {Vermeij, 2014 #1211}. Respiratory waves are generated from a full cycle of respiration and are found between 0.25 Hz – 0.4 Hz {Tong, 2010 #1222}. Cardiac waves are concentrated around 1 Hz and are spatially global and temporally coherent {Tian, 2008 #1214}. The shorter spaced detectors only detect scalp hemodynamics (Fig. 1-2) and can therefore be used as a reference to filter out

physiological hemodynamics contributions to the activation signal as will be explained further in section 2.3. {Tong, 2011 #1236}.

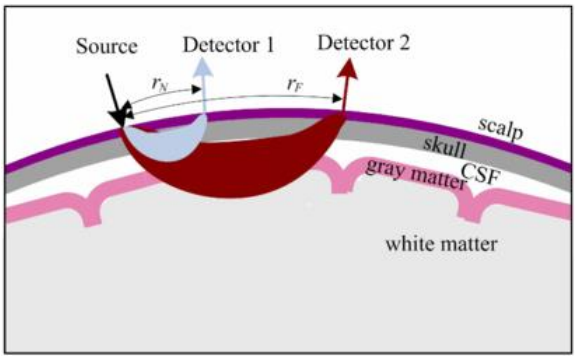

Figure 1-2 Photon trajectory at shorter and farther spaced detectors {Zhang, 2012 #1244}

This optical method for imaging is very applicable in a clinical as well as a lab setting. The advantage of using this method is that it is agile, portable, non-ionizing and relatively low cost. Since the optodes are placed and held along the scalp, it is relatively immune to motion artifacts due to breathing or any involuntary movements {Bunce, 2006 #1234}.

1.2.3 The Optical Window

Optical imaging through biological tissues is challenged by their highly scattering and absorbing properties at optical wavelengths that limit photon penetration {Bunce, 2006 #1234}.Light absorption by blood and tissue components is very high in the green-blue optical and ultraviolet wavelengths, while absorption by water becomes dominant at wavelengths greater than 1200 nm {Bunce, 2006 #1234} {A P Gibson1, 2005 #1247}. In-between these two wavelength regions, the NIR electromagnetic spectrum is termed as the optical window for noninvasive optical imaging since absorption from blood is reduced significantly in this wavelength range (700 – 900 nm) {Bunce, 2006 #1234}. Within the optical window scattering is dominant over absorption, which enables NIR photons entering tissue to scatter multiple times and exit without being absorbed. {A P Gibson1,

2005 #1247}. The absorption spectra for HbO and Hb in the optical window are shown in Figure 1-3.

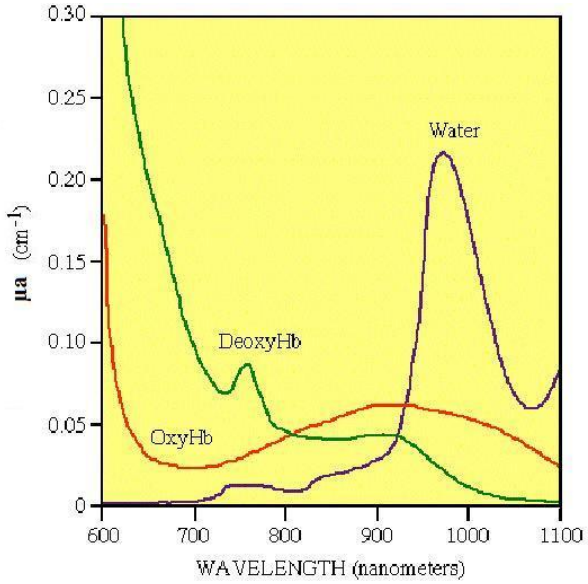

Figure 1-3 Absorption Spectra of HbO, Hb and Water in Optical Window {Bunce, 2006 #1234}

Optical imaging of neuronal function, which is the focus of this work, exploits the optical window that allows NIR photons to reach the cortical surface and detect time-dependent changes in absorption by HbO and Hb occurring due to neurovascular coupling. The NIR light that bounces back to the head surface is detected as a function of time and the areas of activation are localized by an imaging method introduced here below.

1.2.4 Modified Beer-Lambert Law (MBLL)

The optical properties of tissue, absorption and scattering, are wavelength dependent {Bunce, 2006 #1234}. The change in optical density (ΔO.D.) is directly proportion to change in the absorption co-efficient (μ_a), which is the result of change in chromophore concentrations in tissue (Eq. 1.2) {Boas, 2004 #1204}. The intensity of light

detected at the detectors of an fNIRS system is thus a function of change in concentration of HbO and Hb. The MBLL is an empirical formula to quantify this measured light intensity at the surface for a highly scattering medium like biological tissues {Strangman, 2002 #1205}. The formula enables calculation of light attenuation along a scattering medium and has been applied in measuring neuronal activity associated with motor, visual and auditory tasks {Bunce, 2006 #1234}. The formula, as represented in Equation 1.1 enables calculation of change in optical density as a natural logarithm of change in detected light intensity {Montcel, 2005 #1209}. This change in optical density is interpreted as the change absorption in tissue due to drawing of HbO as a result of neuronal activation.

$$O.D = log_{10}(I_0/I) = \epsilon CL = k*C \qquad 1.1$$

O.D = Optical Density

Io = Incident Light Intensity

I = Detected Light Intensity

ϵ = Extinction co-efficient of absorption

C = Concentration of absorption

L = Path length through tissue

K = constant (ϵ*C)

$$\Delta O.D = \Delta \mu_a * L \qquad 1.2$$

The actual physical path length of diffused light through tissue is calculated by scaling the source-detector separation distance by the Differential Path-length Factor (DPF) {Okada, 1997 #1229}. The DPF is a constant based on the optical properties of the

tissue medium. The formula to calculate L is as shown in Equation 1.3, assuming a homogenous medium.

$$L = d*DPF \qquad 1.3$$

d = Source-detector separation distance

DPF = Differential Path length Factor

Equation 1.2 can be extended to in-homogenous medium by considering several homogenous layers and defining Partial Path-length Factor (PPF) per tissue type. {Fukui, 2003 #1228} {Tian, 2008 #1214}.

1.2.5 Tissue Optical Properties

Photons on interaction with tissue organelles undergo elastic and inelastic scattering. With every advancing interaction there is a likelihood of absorption of their energies and deviation in direction of propagation depending on the nature of interacting medium. The Mean Free Path (MFP) is defined as $1/\mu_t$ (transport coefficient); where $\mu_t = \mu_a + \mu_s$ & for biological tissues $\mu_s >> \mu_a$. The absorption coefficient, μ_a, and the scattering coefficient, μ_s are quantification of absorption and scattering events occurring per unit length (unit: mm^{-1}). When a photon undergoes multiple scattering events, the reduced scattering coefficient which is defined as $\mu_s' = \mu_s (1-g)$, is preferred because it takes into account the degree of forward scattering – called the anisotropy factor 'g' {Boas, 2004 #1204} The anisotropy factor 'g' is effectively the mean of cosine of scatter angles and in the range of 0.8 – 1.0 in biological tissues. The Transport Mean Free Path (TMFP) for multiple scattering events is thus defined as $1/\mu_s'$ since $\mu_s' >> \mu_a$ in NIR wavelength of electromagnetic spectrum which results in a diffuse photon flux for distances exceeding a few mm {Ntziachristos, 2010 #1238} {Rolfe, 2000 #1231}. MPF > TMPF {Tuchin, 2007 #1248}. As explained in the section 1.2.1, the source-detector separations for functional

brain imaging need to be several cm, and therefore the reflected light is completely diffused. {A P Gibson1, 2005 #1247}

1.3 Development of FNIRS Applications

Over the years, fNIRS systems have transitioned from simple, single-channel, cumbersome electrode placement systems towards multi-channel, wearable electrode ability systems that are portable. The number of source and detector available for pairing with the instruments are ~32 and can cover the entire head region in adults. Ongoing research is driving fNIRS technology towards low-cost portable set-up {Ferrari, 2012 #1215}.

1.3.1 Limitations to Current Systems

The current fNIRS systems allow about a dozen source-detector pairs and spatial resolution in the range of 1 cm and source-detector separation of 3 cm) {Boas, 2004 #1204} {Bunce, 2006 #1234} {A P Gibson1, 2005 #1247}. The numbers of source-detector pairs that can be used are limited by the detector optode size which is typically 3 mm in diameter (CW-6, Techen Inc., Milford, MA) as well as the cost of increasing the number of channels and associated hardware. Furthermore, the relatively sparse spatial sampling of the head surface by current systems cannot provide any intrinsic imaging of brain anatomical landmarks, which are identified by separate anatomical MRI imaging {Fukui, 2003 #1228} {Boas, 2002 #1208}. The sensitivity of activation detection can also be compromised in fNIRS systems with sparse spatial sampling if activation occurs in areas away from the sensitive detection regions. Finally, studies have described attenuation in light intensity due to interference by hair on the scalp {Orihuela-Espina, 2010 #1221}, though a novel type of brush optode has been proposed as a solution to this problem {Khan, 2011 #1202}.

1.4 Focus of this Work

The focus of this work was to test through simulation studies the feasibility of using a very high source-detector density fNIRS system to resolve some of the problems with existing systems, namely (A) to enable intrinsic identification of anatomical landmarks on the cortical surface and (B) increase activation detection sensitivity irrespective of activation location on the cortex. To that end, simulated fNIRS studies were performed in a realistic head model obtained from anatomical MRI with typical physiological hemodynamic fluctuations added at all tissue depths as explained in the Methods below.

It is shown that very dense spatial sampling with fNIRS can help identify the central sulcus and other nearby sulci based on analyzing baseline hemodynamic fluctuations alone. Furthermore, it is shown that very dense spatial sampling can enhance the detection sensitivity of activation signals very significantly. The capacity of an fNIRS system to identify cortical anatomical landmarks could enable spatial co-registration of images from different patients, or from the same patient on different days, while the increased detection sensitivity could reduce experimental protocol time, or enable performing more activation studies within a given amount of time.

Chapter 2

Methods

2.1 Monte Carlo Simulations Set-up

2.1.1 Monte Carlo Simulations for Photon Migration

Simulations of NIR photon propagation through turbid tissues are typically performed through the diffusion approximations of the photon transport equation {Liemert, 2010 #1220} {A P Gibson1, 2005 #1247} or through Monte Carlo simulations. The diffusion approximations is computationally efficient when simple tissue geometries are assumed, especially the semi-infinite homogeneous medium approximation {A P Gibson1, 2005 #1247} , and has therefore been popular as an approximate method for reconstructing fNIRS activation images {Huppert, 2009 #1207}. Nevertheless, one of the major focus points of this work is the identification of anatomical landmarks on the cortical surface, which implying they study of optical photon propagation in a realistic head geometry, in the presence of spatially heterogeneous background hemodynamic fluctuations. Clearly, a spatially homogeneous model was not appropriate for the scope of this work, and a Monte Carlo approach was selected instead {A P Gibson1, 2005 #1247} {Boas, 2002 #1208} to simulate photon propagation through the head with very high spatial sampling. For the sake of completeness, it should be mentioned that a numerical approach to solving the diffusion equation on a head geometry {Liemert, 2010 #1220} was also briefly pursued initially, but it was found to be very taxing in terms of computer memory requirements to simulate enough spatial detail of the head geometry and was not pursued further.

Monte Carlo algorithms simulate photon propagation stochastically by sampling appropriate probability distributions for distance travelled between two sequential scattering events, and the new travel direction after a scattering event, all the way from

the point of photon entry to its exit from tissue. Source and detector locations are placed on the surface of the head to simulate the spatial arrangement of fibers or fiber bundles used in the fNIRS experiment being simulated. Physiological hemodynamic fluctuations and neuronal activation are simulated by transient changes in photon, as this occurs in real-life experiments too. For the sake of computational efficiency, each tissue voxel with temporally distinct hemodynamic fluctuation patterns is typically tagged with a different index and the total path length of each photon for each tissue index is recorded separately {Boas, 2002 #1208}. Recording photon reflectance data this way enables re-computing the reflectance when the absorption in different voxels changes as a function of time by application of the MBLL (Eq. 1.2) without having to repeat the Monte Carlo simulation {Boas, 2002 #1208} as expressed in Equation 2-1.

$$Reflectance = \Sigma \exp(-\Delta\mu_a.L) \qquad 2.1$$

R = Reflectance in Joules/mm^2

$\sum[\![\Delta\mu_a.L]\!]$ = Sum of product of change in absorption coefficient and path length per photon per tissue type

This way, time-series reflectance data could be generated efficiently. It should also be clarified that the Monte Carlo codes themselves can track the photon transit times and separate them in user-defined time gates in addition to tracking path lengths. In this work all temporal information was added into a single time gate, thus converting the reflectance data to CW. The photon transit times in tissue are in the nanosecond range, which is much shorter than the simulated temporal fluctuations of absorption by blood that occur in the seconds range.

2.1.2 GPU Accelerated Photon Migration

Due to the stochastic nature of Monte Carlo simulations propagating a large number of NIR photons through tissue is very time consuming. This was a major hurdle for the present work, which required the reflectance simulation for a very large number of source-detector pairs separated by 2.5 – 3.5 cm that required a large number of photon histories being simulated so as to attain satisfactory photon statistics at these larger separations (>100 million). The requirements for this work made the use of older Monte Carlo algorithms that were previously used for fNIRS simulations impractical {Boas, 2002 #1208} {LI, 2010 #1224}. Fortunately, there have been recent developments towards speeding up these simulations by exploiting parallel computation ability of Graphical Processing Units (GPUs). We chose Monte Carlo eXtreme (MCX) as the simulation technique {Fang, 2009 #1213} which achieves computation acceleration using parallel threads and low memory latency in a GPU environment. A maximum of 128 tissue types and 3712 detectors for a single source is possible for version 0.9.7 used in this project. MCX results have been validated against the standard diffusion theory solution in a homogenous medium and medium with absorber cases {Fang, 2009 #1213}. The MCX is marked to accelerate by a 300X times when tested on an nVidia GTX 760 graphics card against a conventional Monte Carlo technique, MCVM on a CPU.

With MCX, an nVidia GTX 760 series dedicated graphics card was used to benefit from the time advantage GPU based accelerated processing offered. The output files generated at the end of execution were two – photon history file and flux data file. The photon history file (.mch) traced the history of every photon that reached a defined detector location on the surface and carried information on the path length traversed for each distinct tissue type. It should be repeated that distinct tissue types were not only those with different baseline optical properties (section 2.2 and section 2.3), but also

each tissue voxel that would be assigned distinct time-series fluctuations in absorption depending on the simulation scenarios described here below. The flux data file (.mc2) recorded the flux distribution at every tissue voxel within the tissue model. The voxel units were in millimeters. Three distinct Monte Carlo simulation scenarios were studies in this work:

1. Static Tissue Model: A steady state tissue model with no hemodynamic fluctuations was used at first in order to estimate the minimum number of photons needed to produce a simulation-based standard error that was significantly smaller than typical activation amplitudes as well as perform an initial feasibility test to see if anatomical landmarks can be identified under ideal conditions.
2. Baseline Hemodynamic Fluctuations Model: A tissue model with baseline hemodynamic fluctuations, without any cortical activation, was simulated to assess feasibility of localizing cortical anatomical features with the proposed high density fNIRS system in the presence of physiological noise.
3. Cortical Activation Model: A tissue model with cortical activation in the sensorimotor areas, in addition to baseline hemodynamic fluctuations, was simulated to study how much activation detection sensitivity improves with the proposed high density fNIRS system.

2.2 Static Tissue Model

2.2.1 Anatomical MRI Tissue Model

An anatomical MRI image was obtained from the Statistical Parametric Mapping package (SPM, UCL, www.fil.ion.ucl.ac.uk/spm), which is a freely available tool to process and analyze neuro-imaging data. The software performs spatial tasks such as normalization, re-alignment, eliminating effects from motion artifacts, normalizations, co-registration of functional and anatomical datasets and spatial smoothing on a series of

functional brain images. It is further capable of testing functional hypothesis by statistical methods.

The provided anatomical image was 256 X 256 X108 in voxel dimensions with voxel resolution being 1mm X 1mm X 1.5mm. For the purpose of creating a structure model of the human head, a segmented anatomical image volume was chosen. The regions of cerebrospinal fluid (CSF) merged with the sub-arachnoid space and pia matter, gray matter and white matter regions. The anatomical image was also spatially normalized with the Talairach and Tournox head atlas. (SPM8_manual)). In order to use these images as a tissue model for photon propagation simulations, a few additional image processing steps were necessary. First, it was required to read the Analyze formatted (binary file .img and header file .hdr) files and convert them into a MATLAB (The Mathworks, Inc., Natick, Massachusetts, United States) usable structure. Since SPM was a MATLAB compatible tool, this task was performed with the piece of code that converts each image into a *.mat variable file. (SPM function spm_slice_vol.m). Next, image processing was required to identify and label the various tissue types representing anatomical structures. This 3-D image volume was manually segmented into tissues that were assigned different grey scale values as scalp and skull, which is typically the case for such data sets {Montcel, 2005 #1209}. As MRI pulse sequences are not typically optimized for displaying the skull this appeared to be highly pixilated when labeled based on grey level intensity. Hence a layer of skull and scalp were computationally modeled and added to the MRI image volume by fitting a 3-D spline using the curve fitting toolbox in MATLAB over the top of the pia matter and subarachnoid space to give the model a smooth and even outer layer. The scalp was a 4 mm thick layer and skull was about 5 mm thick {Okada, 1997 #1229}. Once that was achieved, the subarachnoid space and the pia mater were merged with the CSF layer as these are very thin structures

compared to an MRI voxel and their optical properties are not known from current literature. The entire image was considerably voxelated and was smoothened with a 3 X 3 guassian kernel and a 3-D spline interpolant (MATLAB functions: imfilter.m and interp3d.m) to assure that voxelation effects will not have a strong distortion effect on the reflectance measured on the scalp surface. After all the above image processing was performed, the anatomical image was resampled to a 1mm X 1mm X 1mm voxel size in order to reduce possible computational biases in the Monte Carlo simulation that could occur when tissue voxels are a lot larger than the photon transport mean free path. The retained tissue types were tagged by a unique numerical code to assign individual optical properties to them (Table 2-1). The region of the image that was air was tagged as 0.

After identifying and tagging the various tissue types in the brain a 70 mm deep, 70 mm long and 66 mm wide region was chosen around the left sensorimotor region as the anatomical tissue model for simulating photon propagation. To achieve this, the central sulcus that divides the primary motor cortex and somatosensory cortex was used as reference. By the conventional 10-20 electrode system, this region was marked 75 mm from the longitudinal fissure. This structural cropping was done keeping in mind the amount of tissue model size that the simulation could handle while preventing any edge effects in the reflectance results due to photons escaping events at tissue boundaries. Finally the anatomical tissue model was converted to a 70 mm X 70 mm X 66 mm binary file. The transformation is depicted in Figure 2-1.

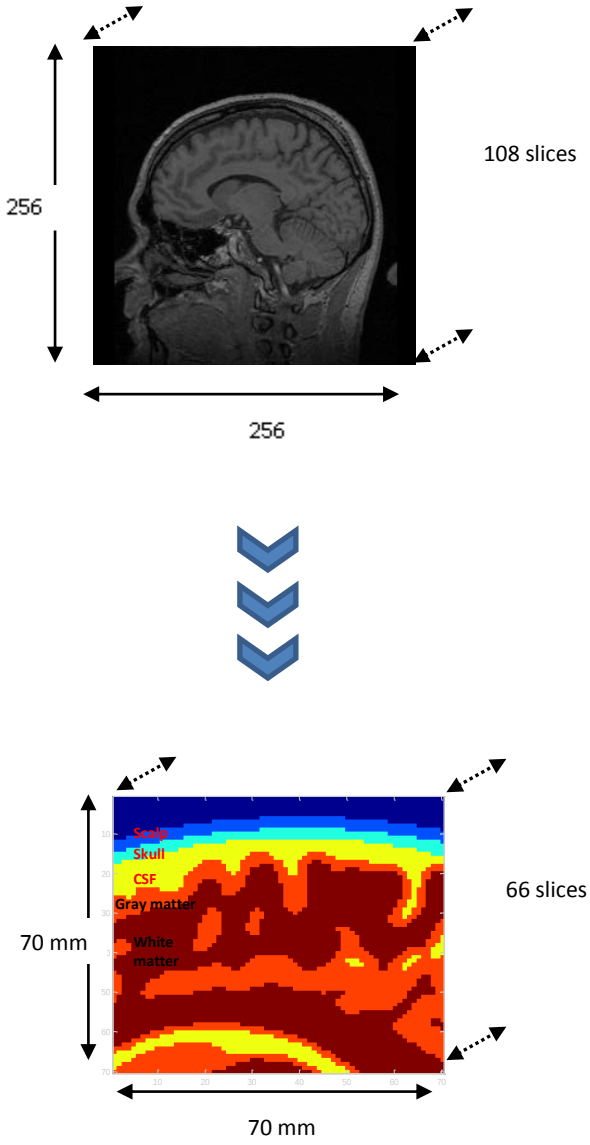

Figure 2-1 Anatomical tissue model from anatomical MRI

2.2.2 Optical Properties of Head and Brain Tissues

The purpose behind identifying the tissue layers with a numerical tag was to assign each tissue with its optical properties in order to prepare the anatomical model for

photon simulation. As previously discussed in 1.2.4 the optical properties were characteristic of the wavelength of light. We chose a commonly used optical neuro-imaging wavelength of 830 nm while assigning optical properties {Okada, 1997 #1229} {Montcel, 2005 #1209}. The units of these were in the millimeter scale for maintaining consistency with the tissue model voxel units.

Table 2-1 Tissue optical properties at 830 nm wavelength

Tissue Layer	Scattering Coefficient (1/mm)	Absorption Coefficient (1/mm)	Anisotropy Factor	Refractive Index
Scalp	9.09	0.060	0.89	1.37
Skull	18.18	0.040	0.89	1.37
Cerebrospinal Fluid (CSF)	0.09	0.001	0.89	1.37
Grey Matter	22.72	0.025	0.89	1.37
White Matter	37.50	0.005	0.84	1.37

Thus a static structural tissue model was created. The source-detector imaging geometry definition was the next step towards setting up inputs for the Monte Carlo simulations for photon migration.

2.2.3 Source-Detector Layout

The simulations required information regarding the source and detector positions indexed to the voxel co-ordinates. A 41 X 20 grid of optodes was placed in a 1 mm grid ensuring the central sulcus to be centrally positioned. The optodes could be selected to function as sources or detectors. One isotropic source perpendicular to scalp was selected per simulation. Since the scalp surface was curved and not flat, the optode placement depth varied along the lateral curvature of brain; albeit the inclination perpendicular to the topmost surface of the voxel. A fiber diameter was set as 500 µm based on the knowledge of perfluorinated fiber dimension tested in the brush optodes

{Khan, 2011 #1202}. By defining the tissue model, the source and detector layout, and explicitly assigning tissue optical properties the human brain model and imaging set-up was ready for simulations.

2.2.4 Data Processing & Analysis

As discussed in section 2.1.2, the photon history output file that the MCX simulations generate contains the pathlengths through each tissue type for each simulated photon reaching a certain detector. Using this information, the fractional photon weight (R) (Eq. 2.1) measured at the detectors is calculated using the MBLL and the summation of all fractional photon weights gives the reflectance that a given detector collects.

The reflectance data were studied as a function of the separation distance between source and detector which dropped steeply with increasing separation. Another dimension to the reflectance measurements was that only detectors at longer separations were the ones that sampled the cortical surface (Fig. 1-2). Thus, as expected, fewer photons emerged at farther detectors. These results were studied for repeatability over five independent Monte Carlo runs, using different initial random number generator seeds, and the deviation from the mean at each detector location was calculated as an estimate of stochastic error from the simulation. The total number of simulated photons was varied until the standard error to the mean (Eq.2.2) was below 1% for source-detector distances up to 35 cm. This number of photo histories of 400 million was kept constant for all subsequent simulations. With the knowledge of baseline hemodynamics and cortical activation fluctuating between 2 - 4 % over the mean optical density, a stochastic variation of less than 1% was assumed safe without extending simulation times excessively {Zhang, 2007 #1240}.

$e = \sigma/\sqrt{n}$ *2.2*

e = Standard error of mean

σ = Standard deviation

n = number of samples

2.3 Baseline Hemodynamic Fluctuations Model

2.3.1 Hemodynamic Fluctuations Simulations

The static tissue model simulations were a preliminary step towards optimizing the simulation set-up and testing the feasibility of using dense fNIRS spatial sampling to detect cortical sulci as will be described below. However, it would be necessary to do so in the presence of hemodynamic fluctuations that are always there, even at rest. In order to achieve that, low frequency oscillations due to physiological fluctuations occurring in the brain during the resting state (section 1.2.2) were included as part of the simulation model.

Simulation of cardiac pulsations was redundant because of their high temporal coherence and spatially global nature. Elimination of cardiac noise from the simulation results is generally straightforward. Simulating Mayer and respiratory waves was necessary because of their lower temporal coherence to create a resting brain model. {Tian, 2008 #1214}. Since fNIRS did not have depth resolution the same noise was added at all depths though the phase of respiration waves across the scalp surface was preserved as recorded. The duration of both the fMRI and fNIRS measurements was adjusted to the same length of 300 sec for consistency.

2.3.1.1 Extracting Mayer waves from an fMRI Data Set

The SPM Epoch data set used for creating the anatomical MRI image also came with 351 fMRI images that were recorded during a visual memory task. The functional MRI images were acquired by T2* - weighted transverse Echo-Planar Imaging (EPI)

method in a descending slice order with BOLD contrast. The image dimensions were 64 X 64 X 24 voxels with voxel resolution being 3 mm X 3 mm X 4.5 mm. The functional image package included 351 images sampled with scan repetition time, TR = 2 s and echo time TE = 40 ms in descending order in Analyze format (binary file .img and header file .hdr). Since the anatomical region of interest in this work was the sensorimotor cortex, which is not affected by the visual memory task, the fMRI over that region were treated as baseline hemodynamic fluctuation data. The fMRI images were sampled at 0.5 Hz temporal resolution and the information in fMRI images was expressed as a Blood Oxygenation Level Dependent (BOLD) metric that was represented by gray scale intensity. The fMRI images were spatially coregistered with the anatomical MRI image and normalized with the Talairach co-ordinates system. The image processing steps performed above for the anatomical MRI image (section 2.2.1) were repeated on the fMRI images. Finally, the images were up-sampled so the voxel size of 1 mm X 1 mm X 1 mm and matched that of the structural tissue model.

By the Nyquist principle of sampling frequency, the fMRI images contained hemodynamic fluctuations in the gray matter region for frequencies less than 0.25 Hz. Thus, in regards to hemodynamic activity they only contained Mayer waves that were slower than respiratory waves. A band pass Butterworth filter of order 3 was applied to isolate the Mayer wave signal in the frequency region of 0.01 Hz – 0.15 Hz. Given that the fMRI data set did not capture respiratory fluctuations or any signal form the scalp (Mayer waves or respiration) it was necessary to obtain these from another source, which was an fNIRS data set.

2.3.1.2 Extracting Baseline Hemodynamic Fluctuations from an fNIRS Data Set

The higher sensitivity of fNIRS enables sampling the hemodynamic signal frequencies that are higher than fMRI. For the analysis performed in this work an fNIRS

data set recorded at 830 nm wavelength from a healthy subject and sampled at 10.35 Hz was procured from the Adaptive Filtering and Depth Compensation Algorithm studies previously by Dr. Fengua Tian. {Tian, 2008 #1214}This dataset included measurements for a NIR brain imager (DyNOT, NIRx Medical Technologies, LLC.,Glen Head, NY) system, in a 3 source and 5 detector probe geometry, where the detectors were located at 1 cm, 2 cm and 3 cm from the sources, along the forehead of a healthy subject. By applying Nyquist sampling theory, this dataset would include hemodynamic fluctuations <5.2 Hz. It was therefore possible to derive respiratory and Mayer hemodynamics from the fNIRS dataset.

Based on the knowledge that detectors closer to source receive photons that traverse the superficial layers and the farther spaced detectors mostly detect photons emerging from deeper layers, the detectors located 1 cm away from the source were chosen as the ones detecting photons primarily traversing through the scalp layer of the head while those detectors at 3 cm away were chosen to detect photons traversing through the deeper cortical layers of tissue. In order to isolate these signals, fNIRS data was filtered with a 3^{rd} order Butterworth band-pass filter and a 9^{th} order Butterworth band-pass filter with lower cut-off frequencies 0.01 Hz, 0.15 Hz, and upper cut-off frequencies 0.25 Hz and 0.4 Hz for Mayer waves and respiratory waves, respectively {Tong, 2010 #1222} {Zhang, 2005 #1241}. Once, that was achieved Δ O.D. was converted into $\Delta\mu_a$ (Eq. 2.3). This conversion was possible by solving for the change in absorption coefficient by the MBLL (Eq. 1.2). The Partial Path length Factor (PPF) is the differential path length factor (DPF) per tissue in the heterogeneous brain model and the information for scalp and gray matter PPF was calculated from the Monte Carlo photon history file for the static anatomical model simulation. The PPF values were determined to be 37.26 and 112.03 for scalp and gray matter respectively.

$$\Delta O.D = \Sigma \Delta \mu_a * PPF \quad 2.3$$

ΔO.D. = total change in optical density

$\Delta\mu_a$ = change in absorption co-efficient

From Equation 2-1 and 2-3 conversion between ΔO.D. and R is a natural logarithmic conversion.

2.3.2 Combining Baseline Hemodynamic Fluctuations from FMRI and FNIRS

At this juncture, we had retrieved Mayer cortical data from fMRI at the voxel level and Mayer and respiratory data for the scalp and cortex as 2D maps from the fNIRS data. To combine this information, the power spectrum of the cortical Mayer from both data sources were compared. A conversion of BOLD signal to $\Delta\mu_a$ conversion constant was determined by comparing power spectrum peaks of Mayer BOLD signal with Mayer ΔO.D for the highest peak at ~0.1Hz. The Mayer wave spectrum from fMRI was then converted to $\Delta\mu_a$ was combined with $\Delta\mu_a$ respiratory wave spectrum from fNIRS. Further, non-Mayer background sampled by fMRI is also combined Mayer and respiratory spectrum and this $\Delta\mu_a$ physiological noise series is added for one voxel of tissue model. As respiratory data is added to the fMRI voxels, a 0.004 sec/mm phase shift laterally away from vasculature along longitudinal fissure and central sulcus is manually added {Filosa, 2007 #1230} {Silverthorn, 2012 #1245}. The phase shift information is calculated based on laterally spaced source-detector pairs from the fNIRS measurement. Amplitude scaling of respiratory and Mayer waves from the fNIRS between cortex and scalp was calculated since unlike gray matter, voxel level hemodynamic information was not available for scalp layer. Based on this amplitude scaling observation, combined noise spectrum from superficial layer of cortex was directly projected onto the scalp. At the end of this process, the tissue model had a 300 sec long baseline hemodynamic fluctuation time-series data

for all scalp and gray matter voxels. Figure 2-2 explains how the filtered noise spectrum from fMRI BOLD data and fNIRS Δμa were sourced. The low and very low frequency oscillations, LFO and VLFO respectively {Obrig, 2000 #1227} {Vermeij, 2014 #1211}, can be distinguished as the two peaks in green at 0.04 Hz and 0.12 Hz. The power spectrum of respiratory waves shown in blue is concentrated around 0.32 Hz. By combining all sources of noise, a single spectrum was created as shown in the bottom panel of Fig. 2-2.

Once the noise information was added, closely synced voxels identified by a cross correlation test with correlation coefficient threshold of 0.5 were grouped together and re-tagged. The purpose of regrouping was to limit the total number of tissue types below the maximum number of 128 that the MCX code allowed. In the final retagged tissue model the gray matter had 12 voxel groups and the scalp had 21, making a total of 33 tissue indices.

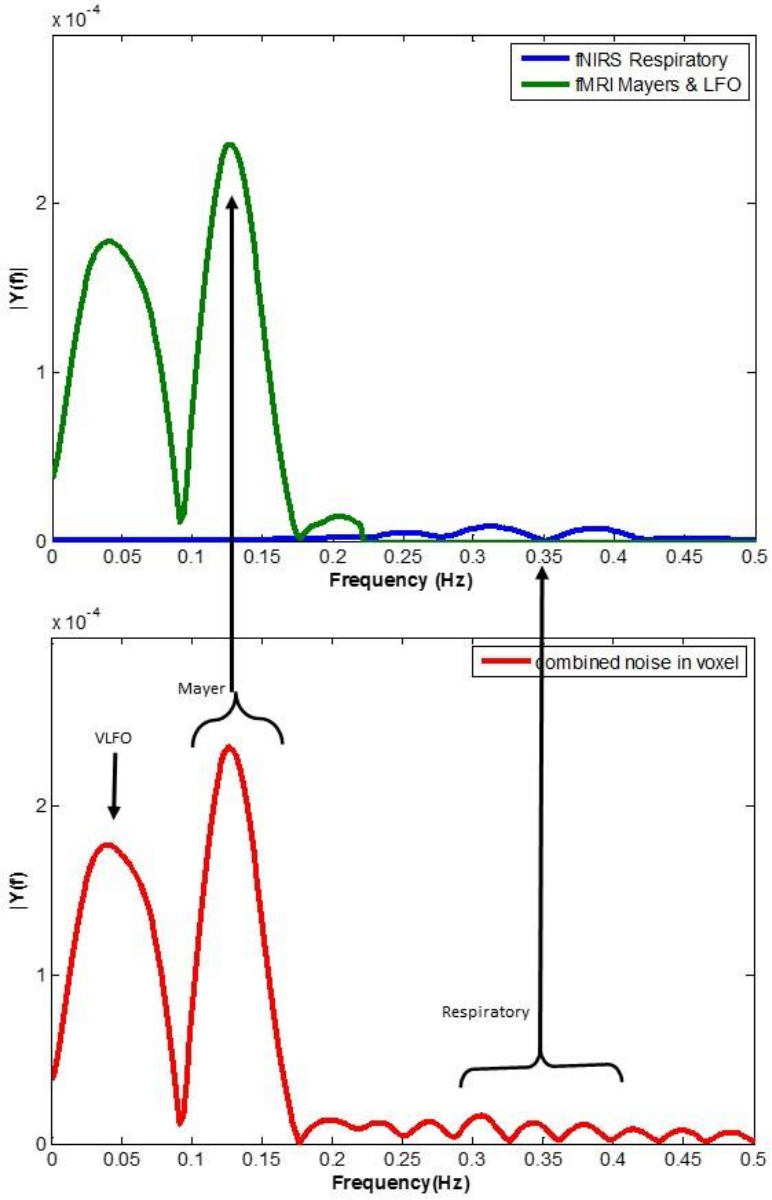

Figure 2-2 (top) Power spectrum for respiratory and Mayer waves from an fNIRS and an fMRI dataset respectively; (bottom) Power spectrum of combined hemodynamic noise embedded into a tissue model voxel

2.3.3 Simulation Set-up for the Tissue Model with Baseline Hemodynamic Fluctuations

The simulation set-up was similar to the static tissue model where 500 µm optodes were placed in a 1 mm grid and each one could be defined as source or detector. A detector grid of 27 X 27 centrally around the central sulcus was selected and 4 simulations were run for 4 sources positioned at every corner of the grid. A square grid was chosen to have a spatially balanced sampling of the central sulcus that was centrally positioned in the field of view (FOV). While defining optical properties of tissue layers, the skull, CSF and white matter had constant optical properties while the scalp voxel and gray matter voxel groups had $\mu_a + \Delta\mu_a$ for time point in the time series data due to the baseline hemodynamic fluctuations added to the background absorption coefficient of each tissue type.

2.3.4 Data Processing & Analysis

Once the simulations were run and output files generated by MCX, the reflectance for the entire time-series data was calculated using the MBLL. Further, noise filtering was essential to remove interference from the hemodynamic fluctuations that were added to the scalp and gray matter areas. These noise sources are complex and cannot be eliminated by simple filtering and this issue has been addressed by some researchers {Tong, 2011 #1236} {Tian, 2008 #1214} {Zhang, 2005 #1241} {Zhang, 2009 #1242} {Saager, 2005 #1233}. Different methods have been adopted to eliminate the effects hemodynamic fluctuations have on reflected photons like the PCA or adaptive noise cancellation {Zhang, 2009 #1242}. We chose the adaptive noise cancellation method by least mean square error technique for removal of Mayer and respiratory complex from the reflectance {Tian, 2008 #1214}.

It is important to note here that the noise removal process was not the exact inverse of the noise (hemodynamic fluctuation) addition process that was performed in

the previous step above. Rather, the physiological noise removal mirrored the method used in physical experiments where a single reference channel at short source-detector distances is used as a reference signal for an adaptive filter algorithm without taking into account and lateral or depth wise variability in these baseline hemodynamic fluctuations.

2.3.4.1 Adaptive Filtering to Remove Hemodynamic Noise

Hemodynamic noise from baseline fluctuations was an integral part of the reflected signal thanks to its globally systemic nature. An adaptive filtering strategy was implemented to remove these background signals {Tian, 2008 #1214}. An adaptive filter tunes its response based on the defined reference signal. In this case, the adaptive filter was applied on detectors beyond 1cm based on the signal measured at detectors at about 1cm distance with respect to the source. By using this strategy the noise originating mostly from the scalp, which was present in all the measurements, was eliminated by a least means square error technique that is at the core of the adaptive filtering algorithm {Khan, 2010 #1200} {Zhang, 2007 #1240}{Zhang, 2009 #1242}. We fine-tuned the filter co-efficients to achieve a clean ΔO.D. response for processing (length $L = 750$ and step size $\mu = 0.001$)

2.3.4.2 Image Reconstruction

Our interest in the baseline reflectance data was to identify anatomical markers on the cortical surface from a resting state simulation data. The image reconstruction technique that is commonly used for reconstructing 2-D topographical map for activation measurements seemed like a good point to start with, although there were a few assumptions to be made due to the nature of set-up of the existing algorithm. Specifically, the image reconstruction algorithm assumes a homogenous infinite medium to calculate A-matrix and solves using the inverse problem. To move ahead, we assumed the simulation model as homogenous and defined optical properties for the hypothesized

model as μa = 0.015 mm^{-1} and μs' = 0.11 mm^{-1}. {Khan, 2011 #1202}. Also, the curvature along the scalp was ignored as the method assumes plane 2-D topography for reconstruction, which is implemented using Equation. 2.4.

$$X = A_{inv} * Y \qquad 2.4$$

X = Reconstructed change in O.D. image (Total Surface Pixels x Time series)

A_{inv} = Inverse of A matrix (Total Surface Pixels x Source-Detector pairs)

Y = Change in O.D. at detector level (Source-Detector pairs x Time series)

'Y' or Change in O.D. at the detectors is required as input for the reconstruction and is deduced from the baseline reflectance data for the four sources using Equation. 2.5:

$$\Delta O.D\ (t) = log_{10}(Reflectance\ /\ mean\ (Reflectance)) \qquad 2.5$$

When combining all possible source-detector pairs to reconstruct an image using Equation. 2.4 the operation was not successful. This was a surprise given that Equation. 2.4 is the standard equation to reconstruct images in the fNIRS field currently. The difference here was that the A-matrix was very large due to the large number of source-detector pairs, which created computer memory problems. Even when the data sizes were reduced the reconstruction results were not successful when using a regular grid source-detector pattern. One preliminary thought is that the large number of source-detector pairs contributes noise that destabilizes the reconstruction. This topic requires further study and was not pursued in this work. It was empirically found though that a ring of detectors at 24 mm away from their respective sources did produce a uniform image

as would be expected for these time-series data. It was this source-detector arrangement that was used in the subsequent image reconstruction step. Once the reconstruction model was set-up, the reconstructed image was analyzed for presence of anatomical markers on the cortical surface by selecting different fNIRS image pixels as seeds and performing a cross-correlation with all other pixels in the image. The premise here was that when the seed pixel happened to be over a sulcus the correlation result with other image pixels would be different form the case where the seed pixel happened to be over a cortical gyrus area.

2.4 Cortical Activation Tissue Model

2.4.1 Simulation Set-up

For the cortical activation model a 35 x 48 detector grid was laid out over scalp on top of the sensorimotor area. The grid dimensions were chosen so that the activation region that was placed at the central sulcus, simulating sensorimotor activation due to finger tapping, was well sampled by many source-detector pairs in the FOV. The detector sizes were 500 μm and the grid was 1 mm dense. A total of 400 million photons were simulated for each source-detector pair.

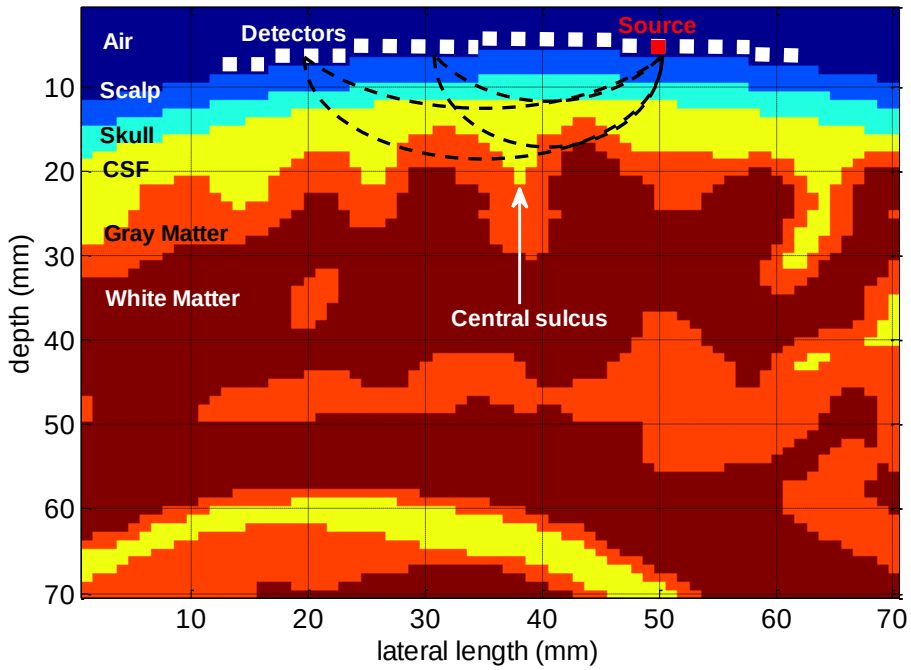

Figure 2-3 Sagittal view of source-detector layout (source = red square; detectors = white squares) and photon trajectories through sensorimotor cortex

2.4.2 Sensorimotor Activation Protocol Simulation

Firstly, the location and size of the activation due finger tapping had to be decided. Based on established homunculus mapping {Silverthorn, 2012 #1245}, the anatomical region that controls or senses motor activation in fingers consists of two regions of approximately 9 mm X 9 mm X 9 mm each and is marked on the anatomical model (Fig. 2-4). The specific dimensions were opted based on fMRI studies that are considered the gold standard in functional imaging {Habermehl, 2012 #1226} {Bunce, 2006 #1234}. Of these two regions, the one anterior to the sulcus was the primary motor cortex and the one posterior was the primary somatosensory cortex.

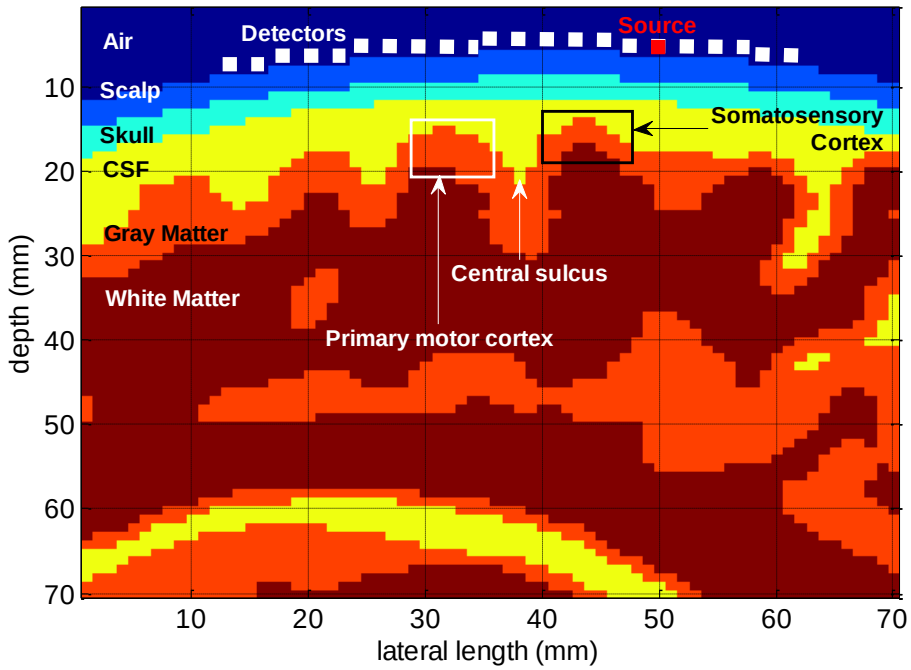

Figure 2-4 Sagittal view of source-detector layout (source = red square; detectors = white squares) and primary motor and somatosensory activation areas along the central sulcus

Secondly, in order to simulate an activation protocol an idealized hemodynamic response to a finger tapping activity was modeled as described below using information from the fMRI literature {Franceschini, 2003 #1206} {Toyoda, 2008 #1237}. Specifically a hemodynamic response function was simulated and convolved with a block-design finger tapping protocol. Specifically, a finger tapping protocol commonly used for fNIRS measurements was chosen to be simulated in the 5 minute long duration of available baseline data. The protocol consists of 16 sec of finger tapping, generally at 1 Hz with 16 sec of no tapping interval before the next tapping epoch. The no-tapping interval is necessary to allow the cortical response to return to baseline before the next activation signal. Seven activation epochs were designed using the non-linear transform for

hemodynamic response function {Toyoda, 2008 #1237}. Also, in this work it was attempted to find the minimum number of taps detectable with the high density sampling fNIRS system. To achieve this latter goal the same block protocol was simulated with a reduced number of taps per block thereby reducing the tapping duration. In the case of short tapping protocols (> 7 sec per block) the non-linearity of neuronal response needs to be taken into account {Toyoda, 2008 #1237}.

In order to use a hemodynamic response function that simulates the non-linearity in neuronal response for short stimulation intervals while also correctly simulating the HRF for longer stimulation intervals a two-step strategy described by Toyoda et. al. {Toyoda, 2008 #1237} was adopted:

In the first step, a neural adaptation function was modeled by Equation. 2.6:

$$n(t) = s(t).\{a.\exp\left(-\frac{t}{b}\right) + 1\} \qquad 2.6$$

t = time after onset of stimulus

n(t) = neural adaptation function

s(t) = stimulus function

a = initial offset constant

b = decay constant

The neural adaptation function was then convolved with an impulse response function to model a BOLD or Hb response to an activation stimulus. The impulse response function is characterized by the Equation:

$$h(t) = \alpha.(\frac{t}{\tau})^{\beta}.\exp(-\frac{t}{\tau}) \qquad 2.7$$

h(t) = impulse response function

α, β, τ are parameters characterizing h(t)

The activation response function r(t) was then characterized by convolving n(t) and h(t) and scaling by a factor $D^{(\varrho-1)}$ {Toyoda, 2008 #1237} {Miller, 2001 #1225} {Soltysik, 2004 #1235}. The constant values were estimated by comparing the response to best-fit match with a standard hemodynamic response function (linear response) by a one-to-one comparison. This brought us to determine a = 3, b = 0.5 and ϱ = 1.14. The free impulse response characterizing parameters α, β, τ were tuned to be 0.05, 9.39 and 4.8 for the 16 sec duration function. In addition to a 16 sec activation protocol, sample protocols were also designed for shorter epoch durations of 1 sec, 2sec, 4 sec and 8 sec for comparing with the conventional 16 sec duration. All protocols maintained a 16 sec interval between two successive tapping events to accommodate the return of cortical activity back to its baseline fluctuations. The free impulse response characterizing parameters were determined in a similar manner for each of the tapping duration. All tapping protocols were designed for 1 Hz tapping frequency and only varied in their length of duration.

The normalized hemodynamic activation response was determined in terms of ΔO.D. The implicit assumption here is that the Hb change that is detected by BOLD had the same time-series profile as the HbO data that are typically reconstructed for fNIRS. The activation amplitude was converted into change in absorption coefficient and scaled by applying a factor of 5% over the maximum deviation in noise amplitude as this is the largest activation amplitude found in the sensorimotor cortex for finger tapping relative to baseline fluctuations {Montcel, 2005 #1209} {Boas, 2004 #1204} {Hoge, 1999 #1217} {Hoge, 2005 #1255} (Fig. 2-5 (top)). This idealized hemodynamic response was added to the previously computed baseline hemodynamic fluctuation time series data to yield a more realistic fNIRS data set (Fig. 2.5 (bottom)).

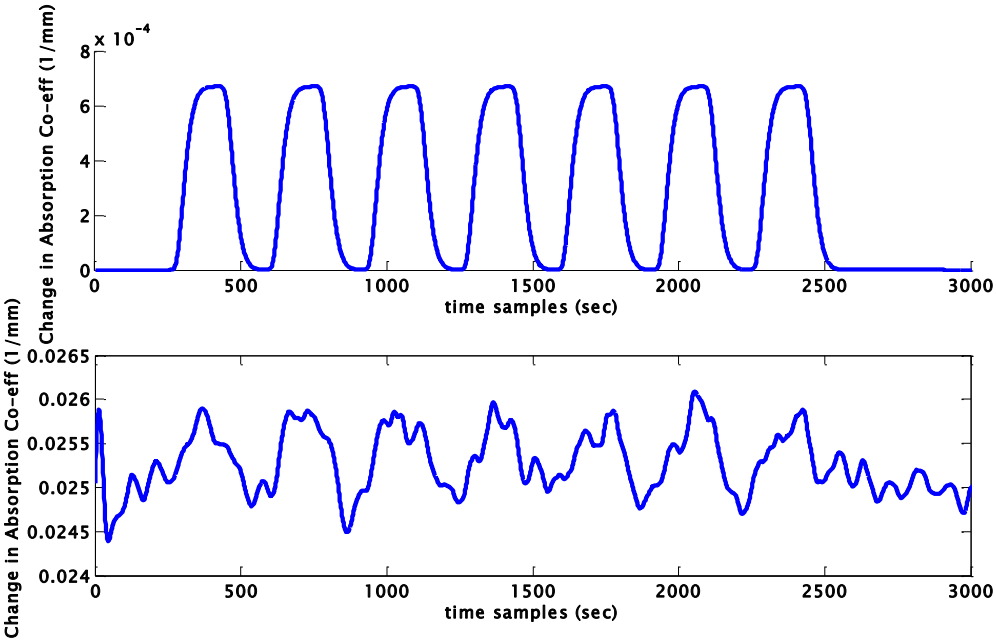

Figure 2-5 (top) Hemodynamic response to a 16 sec finger tapping protocol; (bottom) Activation added over baseline fluctuation data

2.4.3 Data Processing & Analysis

Post simulations, the reflectance results for the sensorimotor activation protocol were calculated and adaptive filtering was performed to remove the hemodynamic noise using a reference form the scalp as done for the baseline fluctuation model described above also. Using the filtered data, the ΔO.D. was calculated similarly to the baseline fluctuation data (Eq. 2.5).

This detector level change in O.D. gave a 2-D representation of the detectors that detect photons emerging from the activation area in the cortex. This information was analyzed further to determine the optimum detector size for maximizing the sensitivity in activation detection relative to existing fNIRS systems.

Chapter 3

Results

3.1 Speed Benchmark for GPU based Monte Carlo

A common MRI brain simulation set-up was run on GPU and non-GPU based systems to test acceleration in simulation speed. The speed benchmarks were as tabulated in Table 3-1. With the right system architecture, MCX is claimed to accelerate processes over 700X fold. {Fang, 2009 #1213} over conventional methods. The MCX was marked to accelerate by a 300X times when tested on an nVidia GTX 760 graphics card against a conventional Monte Carlo technique, MCVM on a CPU. This leap in speed was extremely useful in simulating large number of photon histories in a complex anatomical brain model.

Table 3-1 Performance benchmarks for run-time: 1e8 photons | 30x30 detectors | 5 tissue types | MRI-based brain geometry

Processing Unit	Monte Carlo Method	CUDA cores /Compute Capability	Average Photon Speed (photons/s)
CPU	MCVM	N/A	6 days
Quadro 600	MCX	96 / 2.1	4 hours (35X)
GeForceGTX 760	MCX	1152 / 3.0	30 minutes (300X)

3.2 Monte Carlo Results for Static Anatomical Tissue Model

The static tissue model was tested to optimize the number of photons required for determining reflectance accurately in the 25 – 35 mm source-detector separation range due to the stochastic nature of Monte Carlo methods. In order to determine an adequate number of photons to be simulated, the standard error between 5 independent simulations initiated with different random number generator seeds was calculated. A photon count of 400 million was found to be a good compromise between simulation duration and reflectance accuracy because it maintained the standard error below 1% for separation distances up to 35 mm. The 1 % limit was chosen as acceptable because we estimated a 2 - 4 % variability contribution on addition of physiological noise from hemodynamic fluctuations during the resting state. As the uncertainty in reflectance determination goes approximately with the square root of the number of photon histories simulated, an improvement in uncertainty by a factor of 2 would require running each simulation for 4 times the number of photons. This would be impractically long considering the number of source-detector pair combinations that had to be simulated in this work. Thus, 400 million photons were picked as the optimal number of photons to be simulated for the considered simulation set-up across the various detector layout cases.

The results from the static tissue model simulations were studied by plotting a surface reflectance profile as a function of source-detector separation. As expected, we notice that the reflectance intensity drops very steeply as the separation distance increases. This happens because of the diffuse nature of light and the attenuation of light intensity that occurs during the large number of tissue interactions that occur along the path to surface {A P Gibson1, 2005 #1247} {Fukui, 2003 #1228} {Boas, 2002 #1208} . The primary observation made from the reflectance profile in the static tissue model is the presence of two signal dips in along the reflectance curve near the 20 mm and 25 mm

source-detector separations. Such dips do not occur in the reflectance curves when the underlying medium is homogeneous {Fukui, 2003 #1228} {Boas, 2002 #1208}. Although this finding was exciting as it showed that dense spatial sampling of tissue reflectance could be used to identify cortical landmarks, in real life there are always hemodynamic fluctuations of physiological origin that are always present and could mask these signal dips. In order to assess whither anatomical landmarks could still be identified even in the presence of background hemodynamics further analysis was essential.

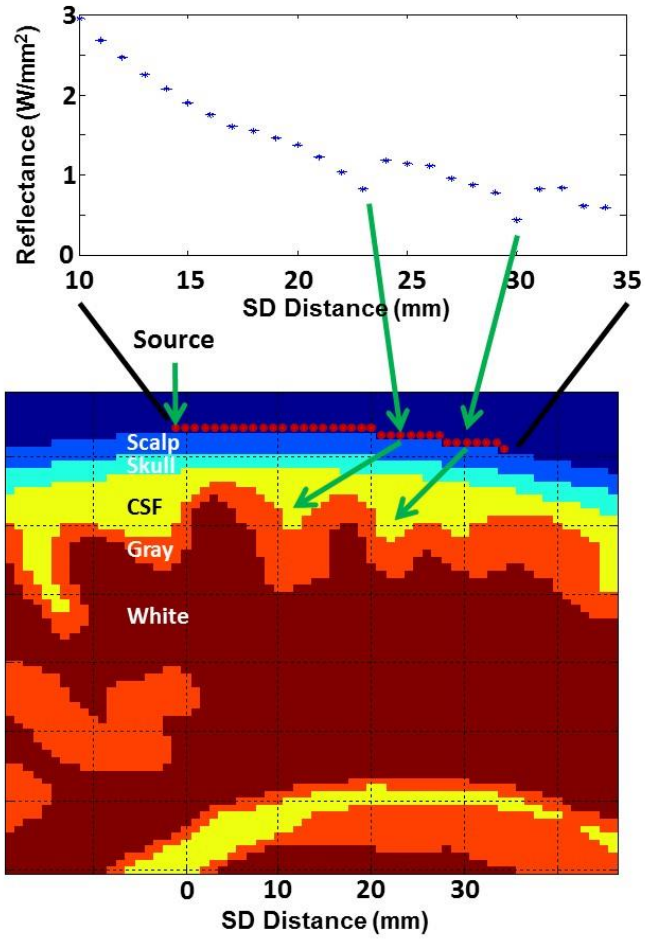

Figure 3-1 (top) Reflectance profile as a function of source-detector separation; (bottom) Sagittal view of tissue model marking the detectors that observe the dips in reflectance

3.3 Monte Carlo Results for a Baseline Hemodynamic Fluctuations Model

A consistent pattern was observed from the cross-correlation analysis of the reconstructed time series when compared between two sulci. The cross-correlation coefficient obtained when time series from the pixels over sulci were typically observed to be > 0.5 and < 0.5 when obtained for pixels over a sulcus and a cortical region. Based on these results, further analyses were performed to substantiate the feasibility of this method to mark anatomical features using the conventional image reconstruction method. The underlying principle of this study was that the correlation seed, when considered as a point on top of a cortical fold, correlates well with data from over other cortical folds, but the same is not true when the seed is selected from cortical plateau areas. Typical results obtained by using this method are shown in Figure 3-2.

The baseline hemodynamic fluctuations simulation model was analyzed by the 2-D image reconstruction algorithm. The results of this method are time-series of $\Delta O.D$ at the topographical level. Since the baseline model was inclusive of hemodynamic noise activity and devoid of any cortical activation, the 2-D reconstruction was investigated for presence of any anatomical markers to support the inferences from the static tissue model results. On cross-correlating data along multiple pixels with the correlation seed on top of a sulcus, we observe other sulci regions correlating well with the seed pixel (Fig. 3-2). For the purpose of testing feasibility of using the cross-correlation technique for anatomical mapping, further analyses treats the correlation coefficient as a metric for sulcus detection with the correlation threshold being 0.5.

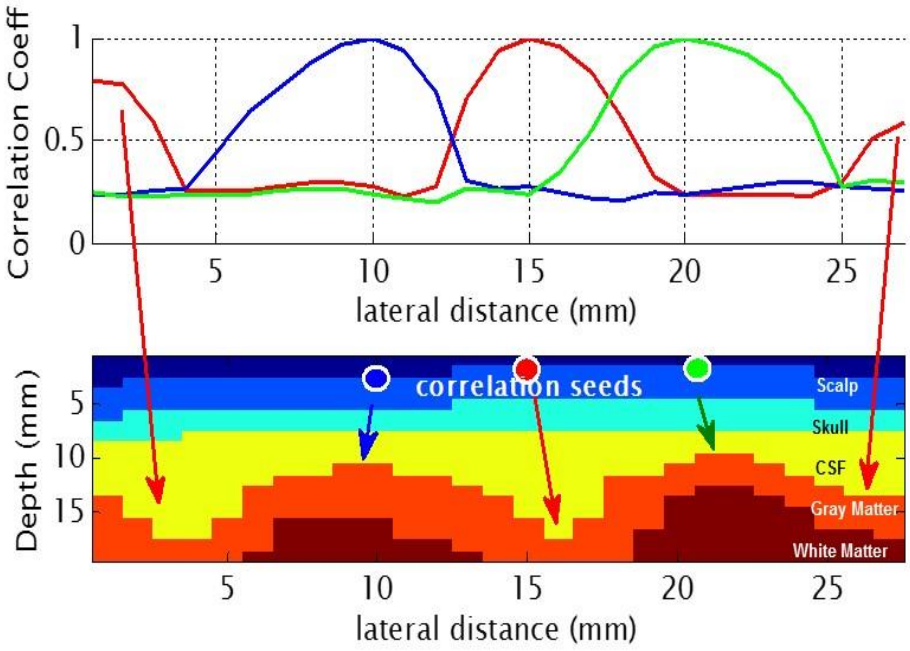

Figure 3-2 (top) Cross-correlation profiles for different correlation seed placements; (bottom) Sagittal view of the tissue model used to verify the correlation trends

3.3.1 Translation of FOV

The imaging FOV was translated laterally so that the central sulcus was moved from the center of the FOV to one side of the FOV in steps of 2mm. The sensitive region of the cross-correlation method was verified by cross-correlating the time-series for the same pixel location in the displaced FOV with relation to the central sulcus. The results of this analysis were interesting, indicating the a slightly displaced seed pixel with respect to the center point of the central sulcus could only detect the central sulcus if it is within 3.5 mm in lateral range, thus making for highly sensitive detection approach. The sensitivity index (cross-correlation coefficient) dropped below 50% at larger distances (Fig. 3-4). This estimation was consistent when the set-up was shifted along different sagittal slices and translating laterally along the brain image volume with the embedded hemodynamic

fluctuations. This finding makes a case that dense spatial sampling would be needed by fNIRS in order to localize anatomical landmarks. The current study had a limitation in that reflectance measurements were only simulated over the part of the sensorimotor cortex where the skull was relatively flat. However, it is possible to set up simulations in future work where source and detector fibers are normal to the local curvature and therefore apply this method to all cortical areas irrespective of head curvature.

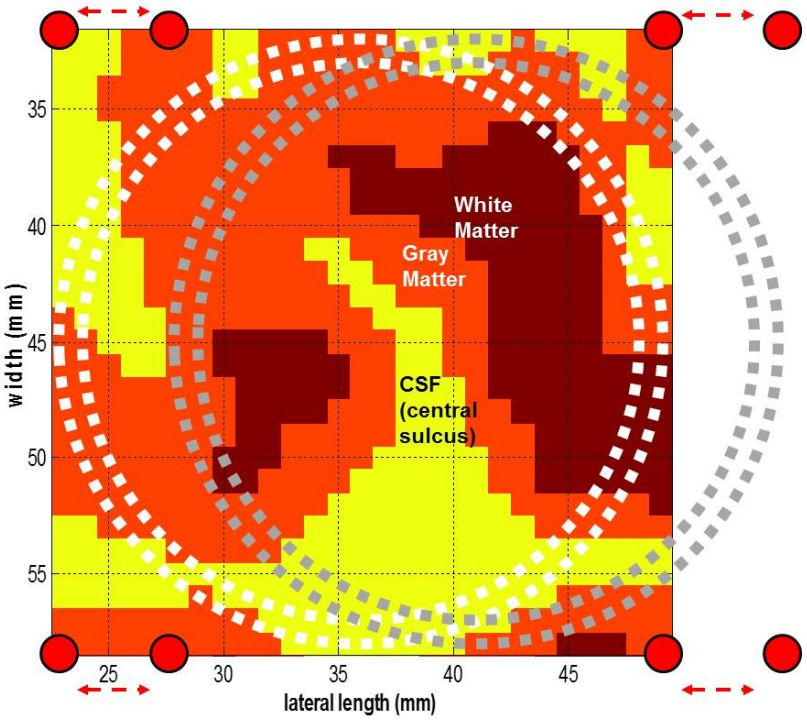

Figure 3-3 Transverse view of tissue model with four sources (red) each paired with 39 detectors (white) translated (white ->grey) laterally

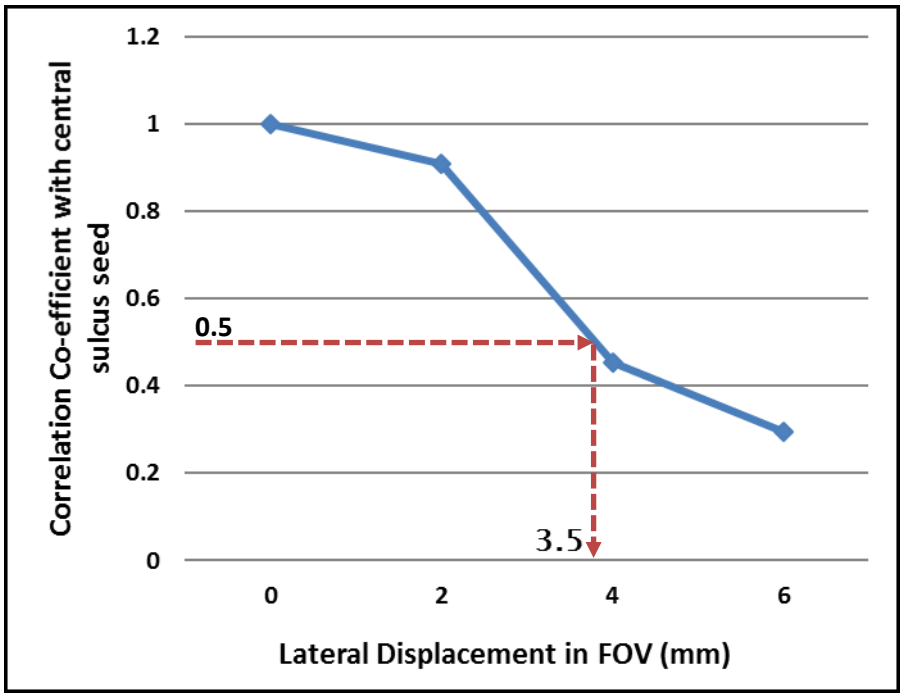

Figure 3-4 Translation of the FOV by 3.5 mm is enough for cross-correlations using the central sulcus as the seed to become decorrelated

3.3.2 Effect of Depth of Sulcus on its Detection Sensitivity

Once the sensitive region within the FOV was estimated, the detection sensitivity with respect to the depth of different sulci on the cortical surface was studied. The previously described cross-correlation technique was used as a metric for analysis of the detection sensitivity with respect to sulcus depth (Eq. 3.1):

Sensitivity = True Positives / Total Number of Positives *3.1*

'True Positives' was defined as correctly determining pixels as sulci by cross-correlation

'Total Number of Positives' was the number of pixels over sulci, which is known from the anatomical image model

Sulci depths were determined as the depth of the CSF column from the deepest point of a fold to the gyri elevation in an adjacent region (Fig. 3-5). Results showed that sensitivity improved for increasing sulci depths (Fig. 3-6) which indicated higher chances of deeper sulci to be detected by this method. The method showed reasonable sensitivity (> 60%) for sulci as shallow as 2 mm which is remarkable. Therefore these simulations showcase the feasibility of the proposed method for anatomically distinguishing cortical folds from cortical plateaus in the presence of resting state hemodynamic fluctuations and thereby enabling the identification of anatomical cortical features by optical means alone.

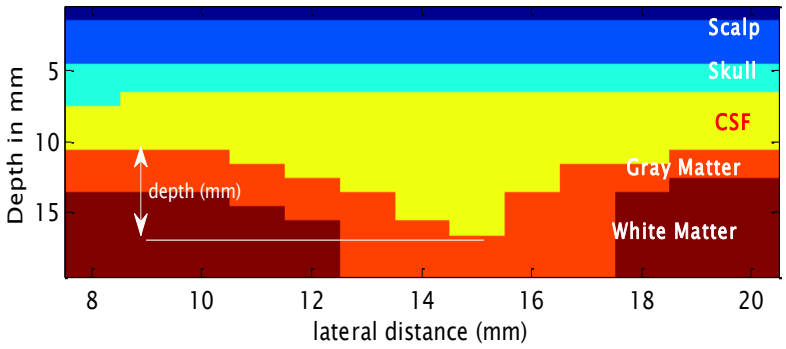

Figure 3-5 Depth measurement along a Sagittal plane of the tissue model

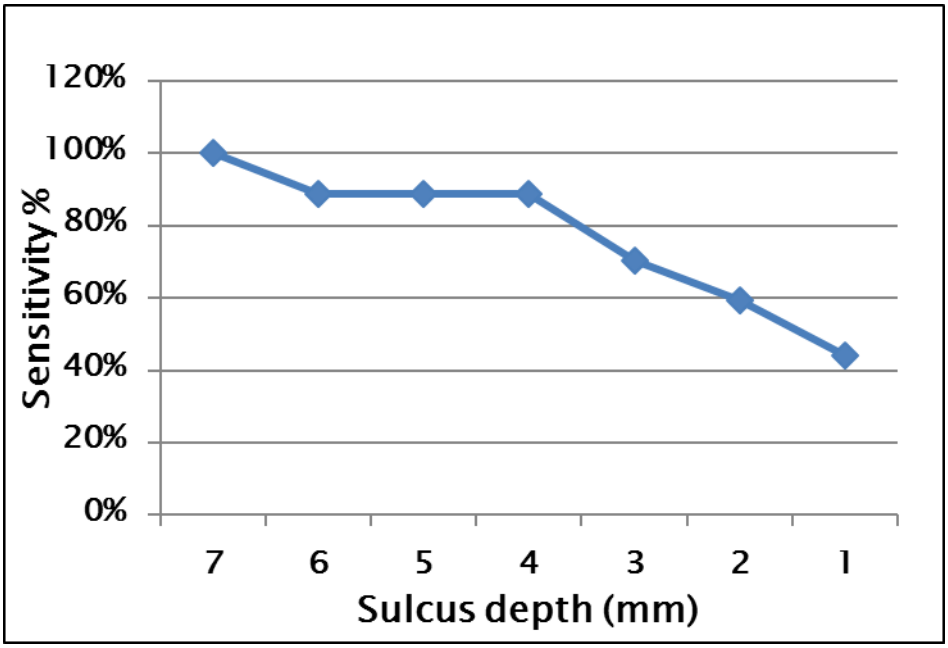

Figure 3-6 Sensitivity of sulcus detection as a function of sulcus depth

3.4 Activation Detection Sensitivity Analysis

By pushing the detector spacing to a tight 1 mm grid layout from an approximate 15 mm spatial sampling that is typical of current fNIRS systems much higher area coverage of the scalp is possible. It is therefore of high interest to calculate the advantages in signal detection efficiency due to dense spatial sampling. In current fNIRS systems detector fiber bundles can often not be packed closer than 10 mm from each other due to the thickness of fiber bundle holders. In the proposed design the fibers are not bundled, but have a brush formation than enables their tight packing on the scalp surface. The proposed system allows grouping the signal read out from multiple detectors virtually and thus attain any desired detector diameter or shape. This dense layout also ensures that there are always detectors positioned at the most sensitive detection region for a given activation location, which cannot be guaranteed with sparser sampling. The optical density spatial profile of the emerging light at the scalp after traversing through the

cortical region of interest can be thought to have a point spread function profile and hence the size and position of the detector becomes crucial for measuring the signal at the most sensitive spot and exploiting the entire measurable signal intensity over the existing background coming from hemodynamics i.e. to obtain a good Signal-to-Noise Ratio (SNR).

The change in optical density at detector level data series determined in section 2.3. (Eq. 2.5) is used towards SNR analysis for investigating the optimum detector size and optimum detector placement with respect to the relative light source position and the activation region location in brain. The SNR is calculated as (Eq. 3.2):

$$SNR = (\text{Mean of amplitude at peak})_{2sec} / (\text{Std. deviation of amplitudes at baseline})_{5sec} \quad 3.2$$

The SNR was calculated for cases where the signal was averaged for multiple epochs. In this case an epoch was defined as tapping for 16 sec at 1Hz and repeating the block 7 times as explained in section 2.4.2. Signal at peak was averaged for a 2 sec interval in order to eliminate noise contributions. Contrast-to-Noise (CNR) is also a commonly used metric to define signal strength over background by cancelling the signal offset that may result from baseline signals. However, with fNIRS time-series data being converted to ΔO.D. the baseline is centered on zero and SNR and CNR yield the same results. Hence, we proceed with the SNR metric to estimate an optimum detector size and detector placement for detecting activation signal and to also quantify by how much the proposed fNIRS architecture could improve the detected activation SNR compared to existing fNIRS systems.

3.4.1 Detector Size Analysis

The physical size of areas of the scalp containing information about activation is larger than the 3 mm fiber bundle detector diameter that current fNIRS systems often

use. The remaining area on the scalp carrying information about activation is not collected in those cases. It was therefore hypothesized that the dense carpeting of the scalp with optical fibers would greatly improve the amount of activation signal detected. However, increasing detector size indefinitely, i.e. integrating signal from an ever larger number of detector fibers, would not improve SNR as beyond a certain size all that is being detected is background hemodynamic noise. This logic led to the hypothesis that an optimal detector size must exist, though at the cost of the spatial resolution of reconstructed images which is not studied in this work. To identify the optimum SNR detector size, we grouped multiple fibers and added their signals within circles of 3 mm, 5 mm, 7 mm, 9 mm, 11 mm, 13 mm, 15 mm, and 17 mm diameter and compared their SNR. These detectors were arranged concentrically and centered over most sensitive area to activation signal, i.e. these were source-detectors whose 'banana'-shaped sensitivity area intersected with the activation region in the brain. The detector layout and SNR results are plotted in Figure 3-7, respectively. SNR increased as the detector size increased until it reached a 13 mm diameter beyond which point it dropped, indicating that background noise contributions were growing stronger for larger diameters. The SNR results were consistent for all the averaged epoch cases. The gain in sensitivity of 13 mm versus the 3 mm diameter detectors was ~125%.

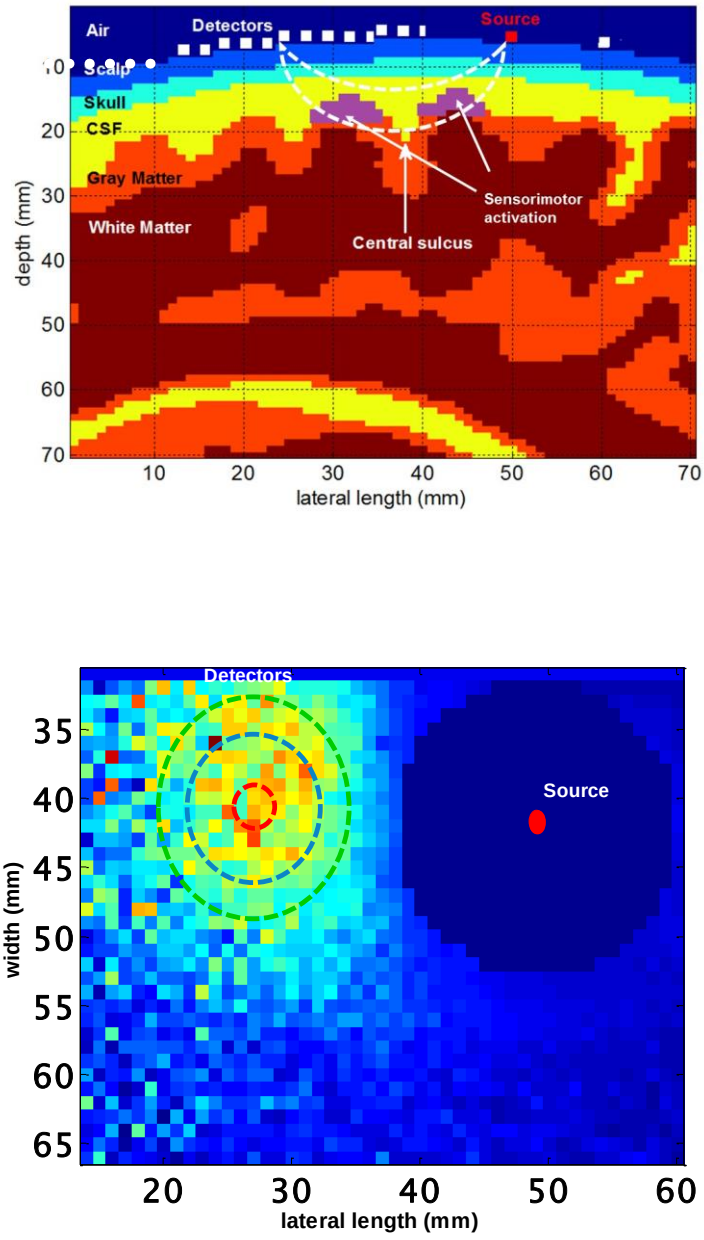

Figure 3-7 (top) Sagittal view of the tissue model with overlaid source, detectors and the 'banana' shaped area that photon trajectories sampled through the activation area (source= red square; detectors= white squares); (bottom) Detector level ΔO.D. map with 2-D

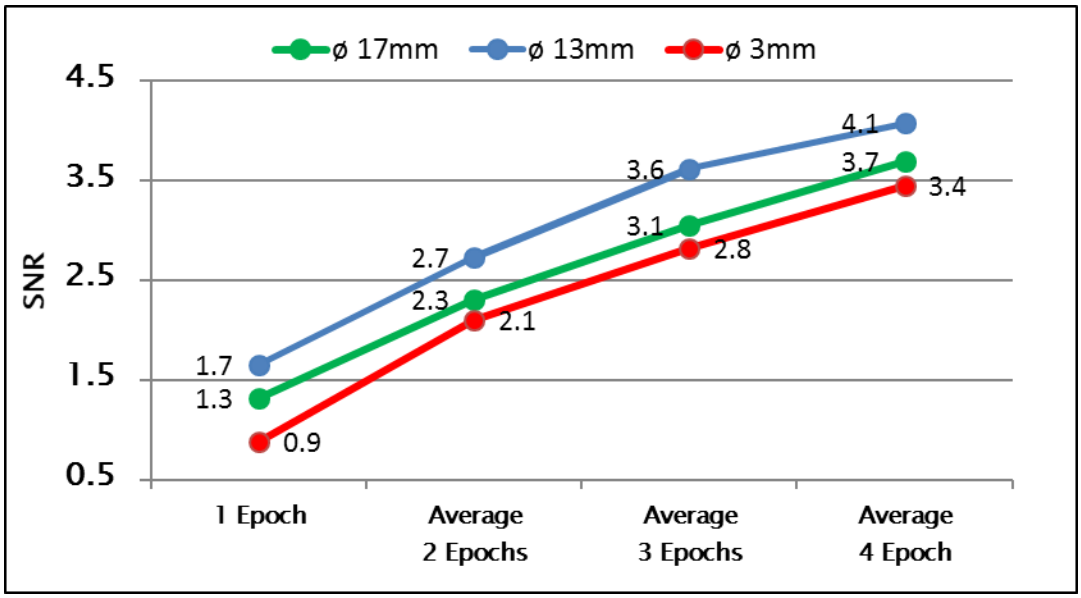

Figure 3-8 SNR results for 1 through 4 averaged 16 sec taping epochs to determine optimum detector size

3.4.2 Detector Placement Analysis

In section 3.3.1, the detectors were intentionally placed over the sensitive detection zone so as to estimate optimum detector size in the best case scenario. In that case the gain in SNR over the standard 3 mm detector was ~125%. However, in reality, such a situation is rare due to the limited spatial sampling permitted by the conventional systems and hence we need to also explore the relative gain in SNR when the traditional 3 mm diameter fiber bundles do not happen to be at the most sensitive location. In this comparison it is implicitly assumed that the dense carpeting with individual detector fibers will always enable to have some detectors centered over the sensitive area. This SNR comparison was performed between 13mm diameter detector optimally positioned (as this is always possible) and the 3mm diameter detectors centered at the source-detector distance by laterally displaced by an angle of 15^{o} and 60^{o} (Fig. 3-9). The13mm detector

placed in line with the light source and the activation region over the most sensitive detectors detected 1.7 – 4.1 times the SNR for a detector displaced at 60^{o} depending on the number of tapping epochs included in the averaging, while the detector deviated by 15^{o} did not show significant additional losses compared to the optical location (Fig. 3-10). The conclusion from this part of the work is therefore that dense spatial sampling can always offer SNR improvements, but the main advantage is that it can do so in a spatially uniform manner that is not possible to attain with the sparse spatial sampling of current fNIRS systems. Further work is needed to determine the SNR at the image, rather than the single source-detector channel level.

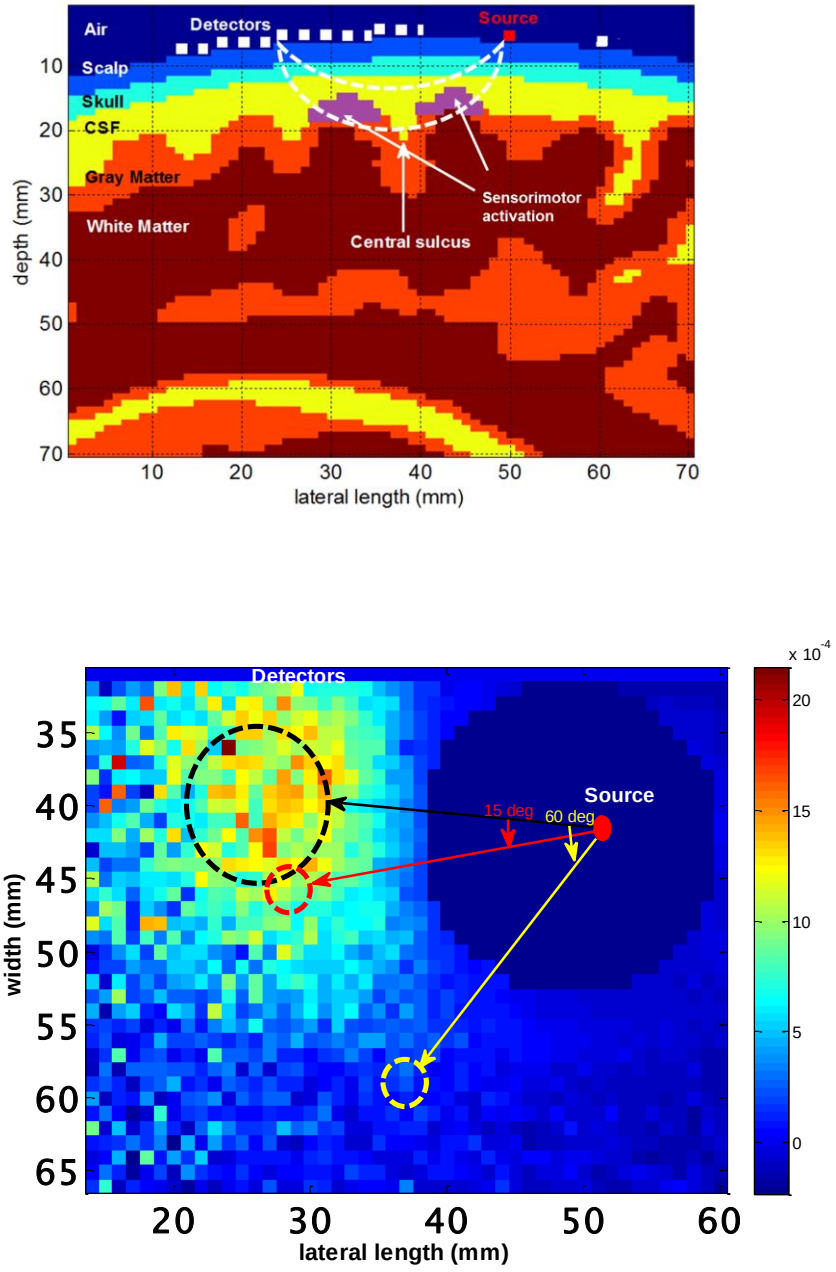

Figure 3-9 Sagittal view of tissue model with overlaid source, detectors and the 'banana' shaped area that photon trajectories sampled through the activation area (source= red square; detector= white squares); (bottom) Detector level ΔO.D. with 2-D detector place

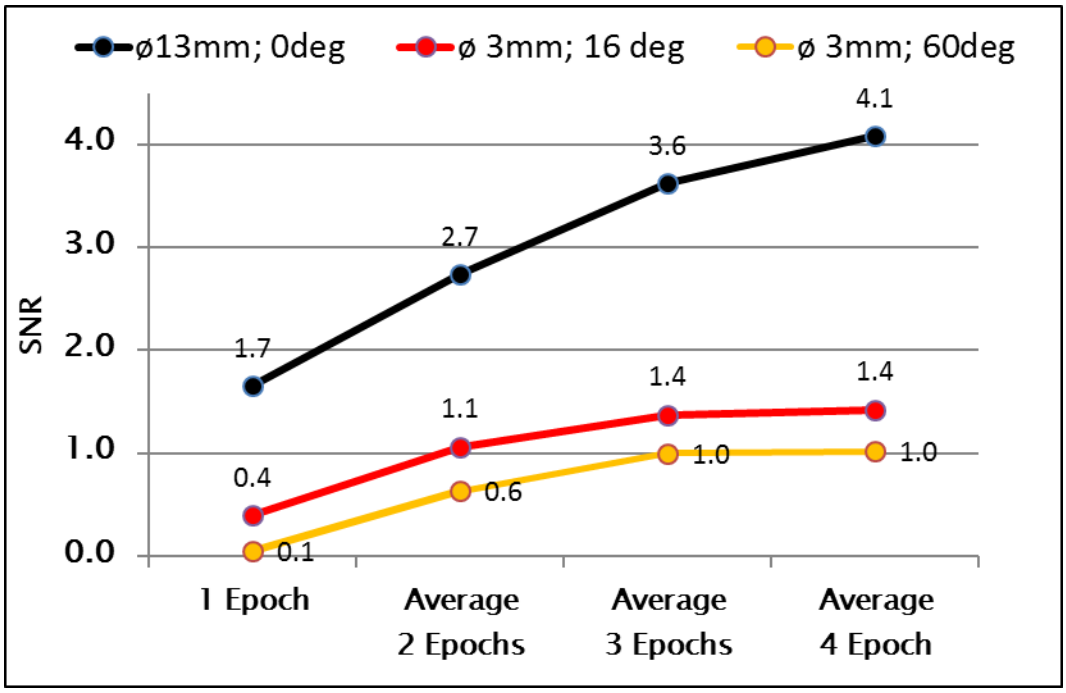

Figure 3-10 Results for 1 through 4 averaged 16 sec tapping epochs were used to quantify how detector placement relative to the activation location affected SNR

3.4.3 Protocol Duration Analyses

Based on the merits in detection sensitivity of the13mm diameter detector over the traditional 3mm diameter detector observed in sections 3.3.1 and 3.3.2, it was intriguing to see if epoch duration had an effect on SNR when comparing the optical to the traditional detector sizes. Two epoch duration cases of 4 sec and 8 sec were compared with the original 16 sec case, with all tapping tasks designed at 1Hz for the 13 mm diameter (Fig. 3-11) and 3 mm diameter (Fig. 3-12) detectors respectively. The detectability limit was defined as SNR=1. Based on the SNR results found, minimum detectable epoch duration for the 13 mm diameter detector was ~8 sec whereas it was >16 sec for the 3 mm diameter detector. These simulations results indicate that total

protocol duration could be shortened by at least a factor of two with the high-density fNIRS system.

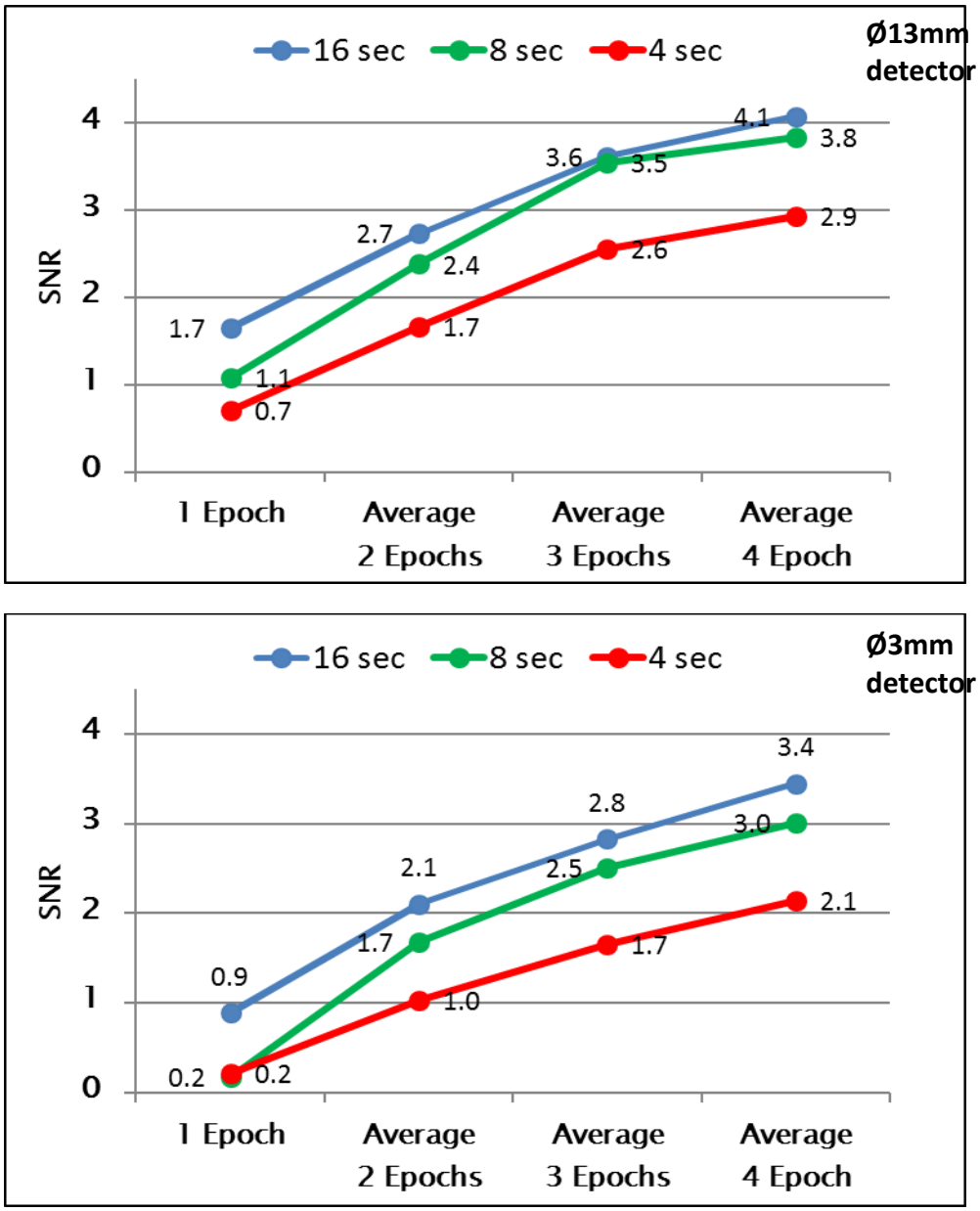

Figure 3-11 SNR results when averaging 1 through 4 epochs for epoch durations of 16 sec, 8 sec, and 4 sec and a 3 mm detector diameter, centered over the activation area on the scalp

3.5 Photon Budget

It was important to estimate the photons that would be reaching the detectors and if the results were true and within the safety limits {Strangman, 2002 #1205}. In order to do so, we calculated the power reaching the detectors and the losses in the path of photons. Tissue propagation losses of the order of 10^6 are commonly observed along the path of light through head. Further, at the interface of scalp with detector fibers, losses might occur commonly and are quantified based on photon detection efficiency {Khan, 2012 #1201}. Losses in photons along the detector fiber occur due to bundling of multiple fibers defined by fiber fill factor (60%) and also due to attenuation along the fiber length (60db/km). Total loss factor assuming good fiber-scalp contact and a 3m long optical bundle is 100% photon detection efficiency X 60% fill factor X 95% attenuation along a 3m long fiber = 57%. Considering safety limits in assuming 5mW power at source the power reaching the Avalanche Photodiode (APD) detector is: 5mW x fraction of photon x total loss factor. Calculating rate of photons reaching the APD for cases of detectors placed at 25mm, 30 mm and 35 mm is shown in Table 3-2.

Table 3-2 Detector photon rate calculation

Source-detector separation distance (mm)	Fraction of photons arriving at detector fibers	Power reaching at APD (5mW x fraction of photon x total loss factor)	Rate of photons at 830 nm (per sec)
25 mm	3.7x10-6	2.6 nW	1×10^{10}
30 mm	1.5x10-6	1 nW	4×10^{9}
35 mm	9x10-7	0.7 nW	3×10^{9}

Since, APD dark count rates are in the range of 500 photons/sec, we conclude that photons reaching the APD are detectable from an individual fiber under laser safety condition for tissues.

Chapter 4

Conclusions and Future Work

4.1 Main Results of the Simulation Studies on the Proposed High Density FNIRS System

This work explored some of the potential improvements that a high density fNIRS system could offer compared to current systems. Though this work did not deal with the great challenges involved in building the hardware for such a system, the simulations performed on a realistic head geometry indicate the great potential of such a system. The proposed capitalizes on the possibility of higher spatial sampling made possible by the advancements in optodes with introduction of the brush optode concept. In this large-scale imaging set-up, detectors can be overlaid on the entire scalp that could offer new capabilities compared to traditional fNIRS systems.

One novel capability that high spatial sampling could offer is the mapping of anatomical features by optical means alone. The 2-D topographical image reconstruction data derived by solving the inverse problem in a resting state condition holds the key for identifying anatomical markers. This was demonstrated by distinguishing a cortical fold from a cortical gyri region by applying the cross-correlation technique described in section 3.3. The resulting sulcus localization was shown to work for signals collected form within 3.5 mm on either side of the region where the sulcus projects on the scalp. Due to the flexibility of the proposed system afforded by dense spatial sampling there would always be some detector fibers within that distance to enable detecting the sulcus. The simulation results also showed that the system could identify even a 2 mm deep sulcus with over 50% sensitivity (section 3.3.2). The proposed analysis method shows potential to tell deeper and shallower sulci apart, but this was not pursued in this work.

The proposed imaging set-up also showed superiority in the SNR detected for cortical activation resulting from a finger tapping motor task when compared to the corresponding performance by a conventional fNIRS system. Results from this study indicated that a detector of diameter size 13 mm was ideal for the strongest activation detection over baseline hemodynamics. The SNR gain ranged from 125% to 400% over the conventional detectors of diameter size 3mm as was calculated based on the placement of the conventional detectors (sections 3.4.1 and 3.4.2). These estimates were made on the premise that the proposed system can group fibers from the region that detects the most activation, which cannot be guaranteed by conventional fNIRS systems due to their much sparser spatial sampling. The estimated improvements in SNR by the proposed system are likely to contribute in reducing the length of protocols, as the improved signal collection efficiency would reduce the minimum number of repetitions needed to attain detectable activation. The proposed system promises a protocol reduction by a factor of ~2 (section 3.4.3) though image based, rather than single channel based studies need to be done to verify this result. Shortening the protocol duration is very important in cases where the subjects have challenges to execute the task and secondarily in reducing the costs associated with administering these measurements.

The simulation results in this study have shown an optimistic potential to take fNIRS imaging studies towards the next generation system that could become a good complement to the other current imaging modalities. More work would however be needed to further assert these computational results and develop the hardware capacities needed to implement such measurements in the real world.

4.2 Future Work

Although the studies performed in this work showed the great potential of a high density fNIRS system to be used as a new technique for marking anatomical features, these results were limited by a few shortcomings.

The most important limitation was using a homogenous ‘A’ matrix (section 2.3.4.2) for the 2-D reconstruction of simulation data for the anatomical tissue model. It would be preferable to calculate the ‘A’ matrix based on the heterogeneous tissues that exist in the brain and assess for any improvements on the anatomical mapping potential of the proposed method. However, calculating an ‘A’ matrix for a heterogeneous tissue geometry by use of Monte Carlo would require very long computation times even when using the MCX code. From the simulation set-up (section 2.3.3) run-time, a prorated time estimate is about 4 months!

The next limitation in this study came from the assumption that the surface curvature in the anatomical tissue model was in fact a horizontal plane projection, to support the existing design of the 2-D reconstruction algorithm (section2.3.4.2). The assumption of a planar scalp surface could have possibly created errors in the computed results for areas near the edge of the FOV used in this work where the curvature was not flat. Clearly this limitation would have to be overcome for the proposed method to be applicable over the entire head surface. However, switching to a curved geometry requires a lot of additional work, e.g. it makes it harder to compute where a point is on a surface or what the distance between two points on the surface is. These challenges would need to be addressed in future work.

Furthermore, when it came to reconstructing images form very densely sampled data, the usual fNIRS image reconstruction algorithm failed due to excessive memory requirements. Furthermore, the number of source-detector pairs that could contribute

more noise than signal into the image reconstruction is a lot higher in this densely sampled geometry. These issues could be addressed by selective filtering of noisy source-detector pairs before data are incorporated into a memory efficient image reconstruction algorithm. (Yodh_DOIcomplex)

Also, image resolution could potentially be improved by improving spatial sampling and signal detection sensitivity. This area of work is called super-resolution and has been explored very little to date. (Zhang_superresolution)

Appendix A

MATLAB Codes for Noise Sourcing and Addition

SOURCING NOISE FROM FNIRS SOURCE

```
clc
% clear all
% close all

%% load input

data = load('WL_830_Fenghua.mat');
fNIRs_data(:,2:16) = data.OD830;
fNIRs_data(:,1) = 0.0966:1/10.35:306.5;
% Time Src1:Det1 Src1:Det2 Src1:Det3 Src1:Det4
Src1:Det5 Src2:Det1 Src2:Det2 Src2:Det3 Src2:Det4
Src2:Det5 Src3:Det1 Src3:Det2 Src3:Det3 Src3:Det4
Src3:Det5
% S3D5 = 1 4 2 5 3 ; S(1 2 3) ; D(1 2 3 4 5)
% we chose Src1:Det4 Src1:Det2 Src1:Det5 Src1:Det3 for analysis
time = fNIRs_data(:,1);
s1_dist1 = fNIRs_data(:,5); s3_dist1 = fNIRs_data(:,16);
s1_dist2 = fNIRs_data(:,3); s3_dist2 = fNIRs_data(:,13);
s1_dist3 = fNIRs_data(:,6); s3_dist3 = fNIRs_data(:,15);
s1_dist4 = fNIRs_data(:,4); s3_dist4 = fNIRs_data(:,12);
figure(3); hold on; plot(time,s1_dist1,'r'); plot(time,s1_dist2,'m');
plot(time,s1_dist3,'g'); plot(time,s1_dist4,'b'); hold off;
```

```
title('Original Reflectance Response @ 830nm Source #1'); legend('1cm' ,'2 cm', '3 cm', '4 cm');
% figure(4); hold on; plot(time,s3_dist1,'r'); plot(time,s3_dist2,'m'); plot(time,s3_dist3,'g'); plot(time,s3_dist4,'b'); hold off;
% title('Original Reflectance Response @ 830nm Source #3'); legend('1cm' ,'2 cm', '3 cm', '4 cm'); axis([100 150 0.2 3]);

%% Band pass filter

fs = abs(1/(time(2,1)-time(1,1)));
tlow = 1;
ylow = -0.05;
thigh = 3172;
yhigh = 0.05;
L = length(time(1:3172));
NFFT = 2^nextpow2(L); % Next power of 2 from length of signal

%% Mayer source #1

[b1,a1] = butter(3,0.15*2/fs);
M_S1_Dist1 = filtfilt(b1,a1,s1_dist1); M_S1_Dist2 = filtfilt(b1,a1,s1_dist2); M_S1_Dist3 = filtfilt(b1,a1,s1_dist3); M_S1_Dist4 = filtfilt(b1,a1,s1_dist4);
[b1,a1] = butter(3,0.01*2/fs,'high');
M_S1_Dist1 = filtfilt(b1,a1,M_S1_Dist1); M_S1_Dist2 = filtfilt(b1,a1,M_S1_Dist2); M_S1_Dist3 = filtfilt(b1,a1,M_S1_Dist3); M_S1_Dist4 = filtfilt(b1,a1,M_S1_Dist4);
```

```
%Plot Time response source #1
figure(13);
plot(time,M_S1_Dist1,'r',time,M_S1_Dist2,'m',time,M_S1_Dist3,'g',time,M_S1_Dist4,'b');
title('Band Filtered Mayer Waves source #1'); legend('1cm' ,'2 cm', '3 cm', '4 cm');
xlabel('Time (sec)'); ylabel('Delta O.D @ 830nm'); %axis([tlow thigh ylow yhigh]);
Y1 = fft(M_S1_Dist1(tlow:thigh),NFFT)/L; Y2 =
fft(M_S1_Dist2(tlow:thigh),NFFT)/L; Y3 = fft(M_S1_Dist3(tlow:thigh),NFFT)/L; Y4 =
fft(M_S1_Dist4(tlow:thigh),NFFT)/L;
f1 = fs/2*linspace(0,1,NFFT/2+1); f2 = fs/2*linspace(0,1,NFFT/2+1); f3 =
fs/2*linspace(0,1,NFFT/2+1); f4 = fs/2*linspace(0,1,NFFT/2+1);
y1 = abs(Y1)/max(abs(Y1));
% Plot single-sided amplitude spectrum source #1
figure(141); subplot(2,1,1);
plot(f1,2*abs(y1(1:NFFT/2+1)),'r');xlim([0 0.2]); %f3,2*abs(Y3(1:NFFT/2+1)),'g');
% divide by 11.1
title('Mayer Single-Sided Spectrum Source #1'); legend('1cm' ,'3 cm');
xlabel('Frequency (Hz)'); ylabel('|Y(f)|'); grid on;

% Plot phase angle source #1
subplot(2,1,2);
phase1 = unwrap(angle(Y1(1:NFFT/2+1))); phase2 =
unwrap(angle(Y2(1:NFFT/2+1))); phase3 = unwrap(angle(Y3(1:NFFT/2+1))); phase4 =
unwrap(angle(Y4(1:NFFT/2+1)));
plot(f1,phase1,'r',f3,phase3,'g');legend('1cm' ,'3 cm'); xlabel('Frequency
(Hz)');ylabel('Phase (Degrees)'); xlim([0 0.2]); grid on
```

```
%% Mayer waves distance comparison pt 1

figure(97); subplot(2,1,1); hold on;
plot(f1,2*abs(Y1(1:NFFT/2+1)),'b'); xlim([0 0.2]);

figure(98); subplot(2,1,1); hold on;
plot(f2,2*abs(Y2(1:NFFT/2+1)),'b'); xlim([0 0.2]);

figure(99); subplot(2,1,1); hold on;
plot(f3,2*abs(Y3(1:NFFT/2+1)),'b'); xlim([0 0.2]);

% Plot phase angle
figure(97); subplot(2,1,2); hold on;
plot(f1,phase1,'b');

figure(98); subplot(2,1,2); hold on;
plot(f2,phase2,'b');

figure(99); subplot(2,1,2); hold on;
plot(f3,phase3,'b');

clear b1 a1 Y1 Y2 Y3 Y4 f1 f2 f3 f4;
%% Mayer source #3
```

```
[b1,a1] = butter(3,0.15*2/fs); % 0.15 Hz Upper limit
M_S3_Dist1 = filtfilt(b1,a1,s3_dist1); M_S3_Dist2 = filtfilt(b1,a1,s3_dist2);
M_S3_Dist3 = filtfilt(b1,a1,s3_dist3); M_S3_Dist4 = filtfilt(b1,a1,s3_dist4);
[b1,a1] = butter(3,0.01*2/fs,'high'); % 0.01 Hz Lower limit
M_S3_Dist1 = filtfilt(b1,a1,M_S3_Dist1); M_S3_Dist2 = filtfilt(b1,a1,M_S3_Dist2);
M_S3_Dist3 = filtfilt(b1,a1,M_S3_Dist3); M_S3_Dist4 = filtfilt(b1,a1,M_S3_Dist4);

%Plot Time response source #3
figure(33);
plot(time,M_S3_Dist1,'r',time,M_S3_Dist2,'m',time,M_S3_Dist3,'g',time,M_S3_Dist4,'b');
title('Band Filtered Mayer Waves source #3'); legend('1cm' ,'2 cm', '3 cm', '4 cm');
xlabel('Time (sec)'); ylabel('Reflectance @ 830nm'); %axis([tlow thigh ylow yhigh]);
Y1 = fft(M_S3_Dist1(tlow:thigh),NFFT)/L; Y2 =
fft(M_S3_Dist2(tlow:thigh),NFFT)/L; Y3 = fft(M_S3_Dist3(tlow:thigh),NFFT)/L; Y4 =
fft(M_S3_Dist4(tlow:thigh),NFFT)/L;
f1 = fs/2*linspace(0,1,NFFT/2+1); f2 = fs/2*linspace(0,1,NFFT/2+1); f3 =
fs/2*linspace(0,1,NFFT/2+1); f4 = fs/2*linspace(0,1,NFFT/2+1);

% Plot single-sided amplitude spectrum source #3
figure(34); subplot(2,1,1);
plot(f1,2*abs(Y1(1:NFFT/2+1)),'r',f3,2*abs(Y3(1:NFFT/2+1)),'g') ; xlim([0 0.2]);
title('Mayer Single-Sided Spectrum Source #3'); legend('1cm' ,'3 cm');
xlabel('Frequency (Hz)'); ylabel('|Y(f)|'); grid on;

% Plot phase angle source #13
```

```
subplot(2,1,2);
phase1 = unwrap(angle(Y1(1:NFFT/2+1))); phase2 =
unwrap(angle(Y2(1:NFFT/2+1))); phase3 = unwrap(angle(Y3(1:NFFT/2+1))); phase4 =
unwrap(angle(Y4(1:NFFT/2+1)));
plot(f1,phase1,'r',f3,phase3,'g');legend('1cm' , '3 cm'); xlabel('Frequency
(Hz)');ylabel('Phase (Degrees)'); xlim([0 0.2]); grid on;

%% Mayer waves distance comparison pt 2

figure(97); subplot(2,1,1); hold on;
plot(f1,2*abs(Y1(1:NFFT/2+1)),'r'); xlim([0 0.2]);
title('Mayer Single-Sided Spectrum for Source-Detectors 1 cm apart');
legend('Source #1 1 cm', 'Source #3 1 cm');xlabel('Frequency (Hz)');
ylabel('|Y(f)|'); grid on;

figure(98); subplot(2,1,1); hold on;
plot(f2,2*abs(Y2(1:NFFT/2+1)),'r'); xlim([0 0.2]);
title('Mayer Single-Sided Spectrum for Source-Detectors 2 cm apart');
legend('Source #1 2 cm', 'Source #3 2 cm');xlabel('Frequency (Hz)');
ylabel('|Y(f)|'); grid on;

figure(99); subplot(2,1,1); hold on;
plot(f3,2*abs(Y3(1:NFFT/2+1)),'r'); xlim([0 0.2]);
title('Mayer Single-Sided Spectrum for Source-Detectors 3 cm apart');
```

```
legend('Source #1 3 cm', 'Source #3 3 cm');xlabel('Frequency (Hz)');
ylabel('|Y(f)|'); grid on;

% Plot phase angle
figure(97); subplot(2,1,2); hold on;
plot(f2,phase2,'r');legend('Source #1 1 cm', 'Source #3 1 cm'); xlabel('Frequency
(Hz)');ylabel('Phase (Degrees)'); xlim([0 0.2]); grid on

figure(98); subplot(2,1,2); hold on;
plot(f1,phase1,'r');legend('Source #1 2 cm', 'Source #3 2 cm'); xlabel('Frequency
(Hz)');ylabel('Phase (Degrees)'); xlim([0 0.2]); grid on

figure(99); subplot(2,1,2); hold on;
plot(f3,phase3,'r');legend('Source #1 3 cm','Source #3 3 cm'); xlabel('Frequency
(Hz)');ylabel('Phase (Degrees)'); xlim([0 0.2]); grid on

clear b1 a1 Y1 Y2 Y3 Y4 f1 f2 f3 f4;

%% Resp source #1

[b2,a2] = butter(9, 0.4*2/fs); % 0.4 Hz Upper limit
R_S1_Dist1 = filtfilt(b2,a2,s1_dist1); R_S1_Dist2 = filtfilt(b2,a2,s1_dist2);
R_S1_Dist3 = filtfilt(b2,a2,s1_dist3); R_S1_Dist4 = filtfilt(b2,a2,s1_dist4);
[b2,a2] = butter(9, 0.25*2/fs,'high'); % 0.25 Hz Lower limit
```

```
R_S1_Dist1 = filtfilt(b2,a2,R_S1_Dist1); R_S1_Dist2 = filtfilt(b2,a2,R_S1_Dist2);
R_S1_Dist3 = filtfilt(b2,a2,R_S1_Dist3); R_S1_Dist4 = filtfilt(b2,a2,R_S1_Dist4);

%Plot Time response
figure(15);
plot(time,R_S1_Dist1,'r',time,R_S1_Dist2,'m',time,R_S1_Dist3,'g',time,R_S1_Dist4,'b');
title('Band Filtered Respiratory Waves for source #1'); legend('1cm' ,'2 cm', '3
cm', '4 cm'); xlabel('Time (sec)'); ylabel('Reflectance @ 830nm'); axis([101 160 -0.005
0.005]);
Y1 = fft(R_S1_Dist1(tlow:thigh),NFFT)/L; Y2 =
fft(R_S1_Dist2(tlow:thigh),NFFT)/L; Y3 = fft(R_S1_Dist3(tlow:thigh),NFFT)/L; Y4 =
fft(R_S1_Dist4(tlow:thigh),NFFT)/L;
f1 = fs/2*linspace(0,1,NFFT/2+1);f2 = fs/2*linspace(0,1,NFFT/2+1);f3 =
fs/2*linspace(0,1,NFFT/2+1);f4 = fs/2*linspace(0,1,NFFT/2+1);

% Plot single-sided amplitude spectrum.
figure(16); subplot(2,1,1);
plot(f1,2*abs(Y1(1:NFFT/2+1)),'r',f3,2*abs(Y3(1:NFFT/2+1)),'g') ; xlim([0.1 0.55]);
title('Respiratory Single-Sided Spectrum for Source #1'); legend('1cm' ,'3 cm');
xlabel('Frequency (Hz)'); ylabel('|Y(f)|'); grid on;

% Plot phase angle
subplot(2,1,2);
```

```
phase1 = unwrap(angle(Y1(1:NFFT/2+1))); phase2 =
unwrap(angle(Y2(1:NFFT/2+1))); phase3 = unwrap(angle(Y3(1:NFFT/2+1))); phase4 =
unwrap(angle(Y4(1:NFFT/2+1)));
plot(f1,phase1,'r',f3,phase3,'g');legend('1cm' ,'3 cm'); xlabel('Frequency
(Hz)');ylabel('Phase (Degrees)');grid on; xlim([0.1 0.55]);

%% Resp waves distance comparison pt 1

figure(87); subplot(2,1,1); hold on;
plot(f1,2*abs(Y1(1:NFFT/2+1)),'b'); xlim([0.1 0.55]);

figure(88); subplot(2,1,1); hold on;
plot(f2,2*abs(Y2(1:NFFT/2+1)),'b'); xlim([0.1 0.55]);

figure(89); subplot(2,1,1); hold on;
plot(f3,2*abs(Y3(1:NFFT/2+1)),'b'); xlim([0.1 0.55]);

% Plot phase angle
figure(87); subplot(2,1,2); hold on;
plot(f1,phase1,'b');  xlim([0.1 0.55]);

figure(88); subplot(2,1,2); hold on;
plot(f2,phase2,'b');  xlim([0.1 0.55]);

figure(89); subplot(2,1,2); hold on;
```

```
plot(f3,phase3,'b'); xlim([0.1 0.55]);

clear b2 a2 Y1 Y2 Y3 Y4 f1 f2 f3 f4;

%% Resp source #3

[b2,a2] = butter(9, 0.4*2/fs);
R_S3_Dist1 = filtfilt(b2,a2,s3_dist1); R_S3_Dist2 = filtfilt(b2,a2,s3_dist2);
R_S3_Dist3 = filtfilt(b2,a2,s3_dist3); R_S3_Dist4 = filtfilt(b2,a2,s3_dist4);
[b2,a2] = butter(9, 0.25*2/fs,'high');
R_S3_Dist1 = filtfilt(b2,a2,R_S3_Dist1); R_S3_Dist2 = filtfilt(b2,a2,R_S3_Dist2);
R_S3_Dist3 = filtfilt(b2,a2,R_S3_Dist3); R_S3_Dist4 = filtfilt(b2,a2,R_S3_Dist4);

% Plot Time response
figure(35);
plot(time,R_S3_Dist1,'r',time,R_S3_Dist2,'m',time,R_S3_Dist3,'g',time,R_S3_Dist4,'b');
title('Band Filtered Respiratory Waves for source #3'); legend('1cm' ,'2 cm', '3
cm', '4 cm'); xlabel('Time (sec)'); ylabel('Reflectance @ 830nm'); axis([101 160 -0.005
0.005]);
Y1 = fft(R_S3_Dist1(tlow:thigh),NFFT)/L; Y2 =
fft(R_S3_Dist2(tlow:thigh),NFFT)/L; Y3 = fft(R_S3_Dist3(tlow:thigh),NFFT)/L; Y4 =
fft(R_S3_Dist4(tlow:thigh),NFFT)/L;
f1 = fs/2*linspace(0,1,NFFT/2+1);f2 = fs/2*linspace(0,1,NFFT/2+1);f3 =
fs/2*linspace(0,1,NFFT/2+1);f4 = fs/2*linspace(0,1,NFFT/2+1);
% Plot single-sided amplitude spectrum.
```

```
figure(36); subplot(2,1,1);
plot(f1,2*abs(Y1(1:NFFT/2+1)),'r',f3,2*abs(Y3(1:NFFT/2+1)),'g') ;xlim([0.1 0.55]);
title('Respiratory Single-Sided Spectrum for Source #3'); legend('1cm' ,'3 cm');
xlabel('Frequency (Hz)'); ylabel('|Y(f)|'); grid on;

% Plot phase angle
subplot(2,1,2);
phase1 = unwrap(angle(Y1(1:NFFT/2+1))); phase2 =
unwrap(angle(Y2(1:NFFT/2+1))); phase3 = unwrap(angle(Y3(1:NFFT/2+1))); phase4 =
unwrap(angle(Y4(1:NFFT/2+1)));
plot(f1,phase1,'r',f3,phase3,'g');legend('1cm' , '3 cm'); xlabel('Frequency
(Hz)');ylabel('Phase (Degrees)');grid on;xlim([0.1 0.55]);

%% Resp waves distance comparison pt 3

figure(87); subplot(2,1,1); hold on;
plot(f1,2*abs(Y1(1:NFFT/2+1)),'r'); xlim([0.1 0.55]);
title('Respiratory Single-Sided Spectrum for sources 1 cm apart');
legend('Source #1 1 cm', 'Source #3 1 cm');xlabel('Frequency (Hz)');
ylabel('|Y(f)|'); grid on;

figure(88); subplot(2,1,1); hold on;
plot(f2,2*abs(Y2(1:NFFT/2+1)),'r'); xlim([0.1 0.55]);
title('Respiratory Single-Sided Spectrum for sources at 2cm apart');
```

```
legend('Source #1 2 cm', 'Source #3 2 cm');xlabel('Frequency (Hz)');
ylabel('|Y(f)|'); grid on;

figure(89); subplot(2,1,1); hold on;
plot(f3,2*abs(Y3(1:NFFT/2+1)),'r'); xlim([0.1 0.55]);
title('Respiratory Single-Sided Spectrum for sources at 3cm apart');
legend('Source #1 3 cm', 'Source #3 3 cm');xlabel('Frequency (Hz)');
ylabel('|Y(f)|'); grid on;

% Plot phase angle
figure(87); subplot(2,1,2); hold on;
plot(f1,phase1,'r');legend('Source #1 1 cm', 'Source #3 1 cm'); xlabel('Frequency
(Hz)');ylabel('Phase (Degrees)'); xlim([0.1 0.55]); grid on

figure(88); subplot(2,1,2); hold on;
plot(f2,phase2,'r');legend('Source #1 2 cm', 'Source #3 2 cm'); xlabel('Frequency
(Hz)');ylabel('Phase (Degrees)'); xlim([0.1 0.55]); grid on

figure(89); subplot(2,1,2); hold on;
plot(f3,phase3,'r');legend('Source #1 3 cm','Source #3 3 cm'); xlabel('Frequency
(Hz)');ylabel('Phase (Degrees)'); xlim([0.1 0.55]); grid on

clear b2 a2 Y1 Y2 Y3 Y4 f1 f2 f3 f4;

%% Reconstruction of scalp signals in time domain
```

```
err1=sqrt(mse(M_S1_Dist3/1.4,M_S1_Dist1));
figure(53); plot(time,M_S1_Dist3/1.4,'r',time,M_S1_Dist1,'g');
title('Reconstructed scalp Mayer Waves source #1'); legend('1cm reconstructed' ,
'1 cm recorded'); xlabel('Time (sec)'); ylabel('O.D @ 830nm'); %axis([tlow thigh ylow
yhigh]);
err2=sqrt(mse(M_S3_Dist3/1.6,M_S3_Dist1));
figure(55); plot(time,M_S3_Dist3/1.6,'r',time,M_S3_Dist1,'g');
title('Reconstructed scalp Mayer Waves source #3'); legend('1cm reconstructed' ,
'1 cm recorded'); xlabel('Time (sec)'); ylabel('O.D @ 830nm'); %axis([tlow thigh ylow
yhigh]);

err3=sqrt(mse(R_S1_Dist3/0.96,R_S1_Dist1));
figure(73); plot(time,R_S1_Dist3/0.96,'r',time,R_S1_Dist1,'g');
title('Reconstructed scalp Respiratory Waves source #1'); legend('1cm
reconstructed' , '1 cm recorded'); xlabel('Time (sec)'); ylabel('O.D @ 830nm'); %axis([101
160 -0.005 0.005]);
err4=sqrt(mse(R_S3_Dist3/0.91,R_S3_Dist1));
figure(75); plot(time,R_S3_Dist3/0.91,'r',time,R_S3_Dist1,'g');
title('Reconstructed scalp Respiratory Waves source #3'); legend('1cm
reconstructed' , '1 cm recorded'); xlabel('Time (sec)'); ylabel('O.D @ 830nm'); %axis([101
160 -0.005 0.005]);

%% Save time series for grey and scalp after verification
```

```
%close all;
save('fNIRs_mayer_resp_filtered_data','fs','time','M_S1_Dist3','M_S1_Dist1','M_S
3_Dist3','M_S3_Dist1','R_S1_Dist3','R_S1_Dist1','R_S3_Dist3','R_S3_Dist1');
%clear all;
```

SOURCE NOISE FROM FMRI SOURCE

```
% % The code takes ~2 hrs for execution so save variables intermittently
% % It does the following : Upscale fMRI data from 3mm*3mm*3mm to
1mm*1mm*1mm
% % It Maps the normalized fMRI data series with bias-corrected anatomical
structure
% % It filters the f-MRI data for Mayer waves and up-samples it from 0.5 Hz
% % to 10.35 Hz to match the f-NIRs data
% % It performs cross-correlation between all grey voxels and based on USER
% % INPUT cross correlation metric groups the grey voxels together
% % It forcefully writes the voxels in premotor activation as independent
% % tissue types
% % It also groups the scalp voxels based on corresponding grey voxels
%
%% f-MRI Input and band pass filtering

clear all;
data = load('fseries.mat');
series = data.img_fseries(6:28,16:40,18:42,1:153); % permute normalized to bias
corrected ref
clear data;

%% Up-scale fMRI data to 1mm*1mm*1mm
```

```
[X,Y,Z]=ndgrid(1:23,1:25,1:25);
[X1,Y1,Z1]=ndgrid(1:1/3:23,1:1/3:25,1:1/3:25);
keep_V2=zeros(66,70,70,153);
for t = 1:size(series,4)
   for z = 1:size(series,3)
      temp_data(:,:,size(series,3)-z+1)=series(:,:,z,t); % reverse z
   end
   V2=interpn(X,Y,Z,temp_data,X1,Y1,Z1,'nearest');
   keep_V2(:,:,:,t) = V2(1:66,2:71,2:71); % keep reqd data in 1mm dimension
   clear data temp_data keep_data V2;
end
clear series X Y Z T X1 Y1 Z1 T1;

%% Anatomical Input

load('tissue_model.mat');
grey = tissue_model==4;
grey_window=zeros(66,70,70,153);
for t = 1:153
   grey_window(:,:,:,t)= grey;
end
scalp = (tissue_model==1);
clear t;
```

```
%% Map f-MRI to anatomical input
fMRI_grey_map = grey_window.*keep_V2;

i = 1;j=1;
for ref_x = 1:66
   for ref_y = 1:70
      for ref_z = 1:70
         if grey(ref_x,ref_y,ref_z)==1
         coords_g(i,1:3) = [ref_x,ref_y,ref_z];
         time_series(i,1:153) = (fMRI_grey_map(ref_x,ref_y,ref_z,1:153));
         i=i+1;
         end
         if scalp(ref_x,ref_y,ref_z)==1
         coords_s(j,1:3) = [ref_x,ref_y,ref_z];
         j = j+1;
         end
      end
   end
end
clear i j ref_x ref_y ref_z grey_window keep_V2;

%% Filter Mayer waves from f-MRI

fs = 0.5; %fMRI frequency
for row = 1:size(time_series,1)
```

```
    tic
    [b2,a2] = butter(3, 0.15*2/fs);
    filt_high(row,1:153) = filtfilt(b2,a2,time_series(row,:));
    [b1,a1] = butter(3, 0.01*2/fs,'high');
    mayer_series(row,1:153) = filtfilt(b1,a1,filt_high(row,:));
    toc
end
clear row fs a2 b2 a1 b1 filt_high;

%% Filter non-Mayer waves from f-MRI for furture inclusion

fs = 0.5; %fMRI frequency
for row = 1:size(time_series,1)
    tic
    [b2,a2] = butter(3, 0.01*2/fs);
    filt_low(row,1:153) = filtfilt(b2,a2,time_series(row,:));
    [b1,a1] = butter(3, 0.15*2/fs,'high');
    non_mayer_series(row,1:153) = filtfilt(b1,a1,filt_low(row,:));
    toc
end
clear row fs a2 b2 a1 b1 filt_high;

%% Save Mayer series
save('fMRI_preproc_part1');
load('fMRI_preproc_part1');
```

```
%
% %% Upsample the mapped grey f-MRI Mayer series
%
% %     javaaddpath(which('MatlabGarbageCollector.jar'))
% %     jheapcl
for row = 1:size(mayer_series,1)
   tic
   t=1:153;
   T=1:1/(2*10.35):153; % go from 0.5 to 10.35 Hz
   mayer_us_series(row,:) = spline(t,mayer_series(row,1:153),T);
   toc
end

clear t row;

%% Upsample the mapped grey f-MRI non Mayer series

%     javaaddpath(which('MatlabGarbageCollector.jar'))
%     jheapcl
for row = 1:size(non_mayer_series,1)
   tic
   t=1:153;
   T=1:1/(2*10.35):153; % go from 0.5 to 10.35 Hz
   non_mayer_us_series(row,:) = spline(t,non_mayer_series(row,1:153),T);
   toc
```

```
end

clear t row;
%% Save upsampled Mayer series
save('fMRI_preproc_part2','mayer_us_series','non_mayer_us_series','-v7.3');
load('fMRI_preproc_part2.mat')
%% Input f-NIRs preprocessed data

load('fNIRs_mayer_resp_filtered_data.mat')
OD_M_G = (M_S1_Dist3(5:3114))'; % we chose to keep 5 min sequence
OD_M_S = (M_S1_Dist1(5:3114))';
OD_R_G = (R_S1_Dist3)'; % dont time limit for phase shift reasons
OD_R_S = (R_S1_Dist1)';

clear M_S1_Dist3 M_S1 Dist1 M_S3_Dist1 M_S3_Dist3 R_S1_Dist3 R_S1 Dist1
R_S3_Dist1 R_S3_Dist3;
% Convert Mayer and Respiratory from change in O.D to change in MuA

Mua_M_S = OD_M_S/36.04;
Mua_M_G = Mua_M_S +(OD_M_G/145.24); % factors derived from mean of
photon path length in tissues from MC simulation
figure(1); plot(Mua_M_S);
Mua_R_S = OD_R_S/36.04;
Mua_R_G = Mua_R_S +(OD_R_G/145.24); % factors derived from mean of
photon path length in tissues from MC simulation
```

```
clear OD_M_G OD_M_S OD_R_G OD_R_S;
%% Normalize f-MRI mayer time series

load('fMRI_preproc_part2.mat')
test_BOLD_M_G = mayer_us_series(47015,5:3114);   % coords_g(47015,1:3) =
(44,37,20)
test_N_BOLD_M_G =
bsxfun(@rdivide,bsxfun(@minus,test_BOLD_M_G,mean(test_BOLD_M_G)),std(test_BO
LD_M_G));%BOLD_M_G(:,1)./norm(BOLD_M_G(:,1));
figure(2); plot(test_N_BOLD_M_G);

clear test_BOLD_M_G;
% Convert BOLD to MuA - Grey & Scalp

factor_BOLD_Mua = max(Mua_M_G)/max(test_N_BOLD_M_G);
N_BOLD_M_G = zeros(size(mayer_us_series(:,5:3114)));
for row = 1:size(mayer_us_series,1)
   N_BOLD_M_G (row,:) =
bsxfun(@rdivide,bsxfun(@minus,mayer_us_series(row,5:3114),mean(mayer_us_series(r
ow,5:3114))),std(mayer_us_series(row,5:3114)));
end
N_BOLD_NM_G = zeros(size(non_mayer_us_series(:,5:3114)));
for row = 1:size(non_mayer_us_series,1)
```

```
            N_BOLD_NM_G (row,:) =
bsxfun(@rdivide,bsxfun(@minus,non_mayer_us_series(row,5:3114),mean(non_mayer_u
s_series(row,5:3114))),std(non_mayer_us_series(row,5:3114)));
        end
        %%
        fs = 10.35%abs(1/(time(2,1)-time(1,1)));
        tlow = 1;
        ylow = -0.05;
        thigh = 3174;
        yhigh = 0.05;
        L = length(time(1:3147));
        NFFT = 2^nextpow2(L);
        f1 = fs/2*linspace(0,1,NFFT/2+1);
        Y1 = fft(mayer_us_series(tlow:thigh),NFFT)/L;
        Y2 = (abs(Y1) - mean(abs(Y1)))/std(abs(Y1));

        figure(14); subplot(2,1,1);
        plot(f1,2*abs(Y2(1:NFFT/2+1)),'r'); xlim([0 0.2]); % divide by 11.1
        title('Mayer Single-Sided Spectrum Source #1'); legend('BOLD');
xlabel('Frequency');
        %%
        % Mayer
        Anat_Mua_M_G = N_BOLD_M_G*factor_BOLD_Mua;
        Anat_Mua_M_S = zeros(size(coords_s,1),size(Anat_Mua_M_G,2));
        for row = 1:size(coords_s,1)
```

```
    row_ref = strmatch(coords_s(row,1:2),coords_g(:,1:2));
    if size(row_ref,1)==0
        Anat_Mua_M_S(row,:) = (1/1.5)*Anat_Mua_M_G(1,:); % unmapped scalp -
grey mapped to first voxel
    else
        Anat_Mua_M_S(row,:) = (1/1.5)*Anat_Mua_M_G(row_ref(1),:);
    end
end

%Non-Mayer
Anat_Mua_NM_G = N_BOLD_NM_G*factor_BOLD_Mua;

% Respiratory

load('delay_matrix.mat')
% the delay file is created in excel keeping 44,37,20 as ref
% the scalp allows a delay of 0.004 sec/mm. Data sampling is 0.0966 sec
% the propagation is away from the partition of the hemisphere and sulcus

Anat_Mua_R_G = zeros(size(Anat_Mua_M_G));
for row = 1:size(coords_g,1)
    Anat_Mua_R_G(row,:) = Mua_R_G(5:3114);
end
Anat_Mua_R_S = zeros(size(Anat_Mua_M_S));
for row = 1:size(coords_s,1)
```

```
    x = coords_s(row,1); y = coords_s(row,2);
    shift_delay = shift(x,y);
    Anat_Mua_R_S(row,:) = Mua_R_S(5+shift_delay:3114+shift_delay);
end

clear row x y shift_delay test_BOLD_M_G test_N_BOLD_M_G N_BOLD_M_G
Mua_R_G Mua_R_S Mua_M_G Mua_M_S;
% Total Mua

Anat_Mua_G = Anat_Mua_M_G + Anat_Mua_R_G + Anat_Mua_NM_G;
Anat_Mua_S = Anat_Mua_M_S + Anat_Mua_R_S;

clear Anat_Mua_M_G Anat_Mua_M_S Anat_Mua_R_G Anat_Mua_R_S;

% Save Delta MuA data
save('Delta_MuA_part_1','Anat_Mua_G','Anat_Mua_S','-v7.3');
%% Group f-MRI grey voxels & scalp voxels based on correlation metric

tag_value_g = 0;
[g,iga,igc] = unique(Anat_Mua_G(:,1:311),'rows');
tag_g = zeros(size(g,1),1);
for row = 1:size(g,1)
    tic
    if tag_g(row) == 0;
        tag_value_g = tag_value_g+1;
```

```
            tag_g(row) = tag_value_g;
            Gcomp(:,1) = gpuArray(g(row,1:311)); % use only 0.5 min sequence
            for j = row+1:size(g,1)
                if tag_g(j) == 0;
                    Gcomp(:,2) = gpuArray(g(j,1:311));
                    Gtest = xcorr(Gcomp(:,1),Gcomp(:,2),'coeff');
                    [argvalue,argmax] = max(gather(Gtest));
                    if argvalue >= 0.5 && argmax >=261 && argmax <=361 % 50% chosen
because it gives an ideal tissue type number
                        tag_g(j) = tag_value_g;
                    end
                end
            end
        end
        toc
    end
    clear row j Gcomp argvalue argmax Gtest;

    tag_value_s = 0;
    [s,isa,isc] = unique(Anat_Mua_S(:,1:311),'rows');
    tag_s = zeros(size(s,1),1);
    for row = 1:size(s,1)
        tic
        if tag_s(row) ==0;
            tag_value_s = tag_value_s+1;
```

```
            tag_s(row) = tag_value_s;
            Gcomp(:,1) = gpuArray(s(row,1:311)); % use only 0.5 min sequence
            for j = row+1:size(s,1)
                if tag_s(j) == 0;
                    Gcomp(:,2) = gpuArray(s(j,1:311));
                    Gtest = xcorr(Gcomp(:,1),Gcomp(:,2),'coeff');
                    [argvalue,argmax] = max(gather(Gtest));
                    if argvalue >= 0.5 && argmax >=261 && argmax <=361 % 50% chosen
because it gives an ideal tissue type number
                        tag_s(j) = tag_value_s;
                    end
                end
            end
        end

        toc
    end
    clear row j Gcomp argvalue argmax Gtest;

    save('Delta_MuA_part_2','g','iga','igc','tag_g','s','isa','isc','tag_s');

    % Delta MuA for unique tags
    [unique_tag_g,uga,ugc] = unique(tag_g,'rows');
    [unique_tag_s,usa,usc] = unique(tag_s,'rows');
    unique_Anat_Mua_G = Anat_Mua_G(iga(uga,1),:);
```

```
unique_Anat_Mua_S = Anat_Mua_S(isa(usa,1),:);
save('Delta_MuA_part_3','unique_tag_g','uga','ugc','unique_tag_s','usa','usc','uniq
ue_Anat_Mua_G','unique_Anat_Mua_S','g','s');

% Grey Scalp Group tagging | Sulcus Forced tagging
% re-rank = 1 Skull; 2 CSF; 3 White; 4:max grey Grey;
% max grey:max scalp Scalp; Rest forced sulcus tags
load('tissue_model.mat');
skull_mask = (tissue_model ==2);
csf_mask = (tissue_model ==3);
white_mask = (tissue_model ==5);
tagged_tissue_model = skull_mask*1 + csf_mask*2 + white_mask*3;

offset = 3; % scalp
for i = 1:size(coords_s,1)
   tagged_tissue_model(coords_s(i,1),coords_s(i,2),coords_s(i,3)) =
tag_s(isc(i))+offset;
end

offset = offset+tag_value_s; % grey
for i = 1:size(coords_g,1)
   tagged_tissue_model(coords_g(i,1),coords_g(i,2),coords_g(i,3)) =
tag_g(igc(i))+offset;
end
```

```
        offset = offset + tag_value_g + 1;  % sulcus pre-motor cortex
        tagged_tissue_model(42,31:34,20) = offset; tagged_tissue_model(43,32:36,20) =
offset; tagged_tissue_model(44,33:37,20) = offset;
        tagged_tissue_model(45,36:37,20) = offset;
        tagged_tissue_model(41,29:33,19) = offset; tagged_tissue_model(42,30:34,19) =
offset; tagged_tissue_model(43,30:35,19) = offset;
        tagged_tissue_model(44,30:36,19) = offset; tagged_tissue_model(45,35:37,19) =
offset;
        tagged_tissue_model(38,27:30,18) = offset; tagged_tissue_model(39,27:31,18) =
offset; tagged_tissue_model(40,27:32,18) = offset;
        tagged_tissue_model(41,27:32,18) = offset; tagged_tissue_model(42,27:33,18) =
offset; tagged_tissue_model(43,28:34,18) = offset;
        tagged_tissue_model(44,29:36,18) = offset; tagged_tissue_model(45,35:36,18) =
offset;
        tagged_tissue_model(38,27:29,17) = offset; tagged_tissue_model(39,28:29,17) =
offset; tagged_tissue_model(40,28:30,17) = offset;
        tagged_tissue_model(41,28:32,17) = offset; tagged_tissue_model(42,28:32,17) =
offset; tagged_tissue_model(43,28:34,17) = offset;
        tagged_tissue_model(44,28:35,17) = offset; tagged_tissue_model(45,29:36,17) =
offset;
        tagged_tissue_model(41,30,16) = offset; tagged_tissue_model(42,29:31,16) =
offset; tagged_tissue_model(43,29:32,16) = offset;
        tagged_tissue_model(44,29:34,16) = offset; tagged_tissue_model(45,29:34,16) =
offset;
```

tagged_tissue_model(44,31:32,15) = offset; tagged_tissue_model(45,31:33,15) = offset; tagged_tissue_model(46,31:35,15) = offset;

```
        tagged_tissue_model(48,32:33,14) = offset; tagged_tissue_model(49,32,14) =
offset;

        offset = offset + 1;    % sulcus somatosensory cortex

        tagged_tissue_model(42,37:40,20) = offset; tagged_tissue_model(43,38:41,20) =
offset; tagged_tissue_model(44,39:41,20) = offset;
        tagged_tissue_model(45,40:41,20) = offset;
        tagged_tissue_model(41,36:40,19) = offset; tagged_tissue_model(42,37:41,19) =
offset; tagged_tissue_model(43,38:41,19) = offset;
        tagged_tissue_model(44,40:41,19) = offset; tagged_tissue_model(45,40:41,19) =
offset;
        tagged_tissue_model(38,32:38,18) = offset; tagged_tissue_model(39,33:38,18) =
offset; tagged_tissue_model(40,34:40,18) = offset;
        tagged_tissue_model(41,36:41,18) = offset; tagged_tissue_model(42,37:41,18) =
offset; tagged_tissue_model(43,39:41,18) = offset;
        tagged_tissue_model(44,40:41,18) = offset; tagged_tissue_model(45,40:41,18) =
offset;
        tagged_tissue_model(37,33:39,17) = offset; tagged_tissue_model(38,34:39,17) =
offset; tagged_tissue_model(39,34:39,17) = offset;
        tagged_tissue_model(40,35:40,17) = offset; tagged_tissue_model(41,36:42,17) =
offset; tagged_tissue_model(42,38:42,17) = offset;
```

```
        tagged_tissue_model(43,39:42,17) = offset; tagged_tissue_model(44,40:42,17) =
offset; tagged_tissue_model(45,40:42,17) = offset;
        tagged_tissue_model(38,35:37,16) = offset; tagged_tissue_model(39,35:39,16) =
offset; tagged_tissue_model(40,36:40,16) = offset;
        tagged_tissue_model(41,37:42,16) = offset; tagged_tissue_model(42,39:43,16) =
offset; tagged_tissue_model(43,40:43,16) = offset;
        tagged_tissue_model(44,40:42,16) = offset; tagged_tissue_model(45,41:42,16) =
offset;
        tagged_tissue_model(42,40:42,15) = offset; tagged_tissue_model(43,41:43,15) =
offset; tagged_tissue_model(44,42:44,15) = offset;
        tagged_tissue_model(45,42:44,15) = offset; tagged_tissue_model(46,43:45,15) =
offset;
        tagged_tissue_model(42,42:44,14) = offset; tagged_tissue_model(43,42:44,14) =
offset; tagged_tissue_model(44,42:44,14) = offset;

        clear skull_mask csf_mask white_mask offset;
        save('tagged_tissue_model','tagged_tissue_model');
```

Appendix B

MATLAB Codes for Simulation Input, MCX Simulation, Output post-processing and Analysis

```
DEFINING SIMULATION INPUTS
clc; clear all;
load tagged_tissue_model_new.mat;

%% create binary tissue model in uchar format
fid=fopen('tagged_tissue_model.bin','wb');
bb=fwrite(fid,tagged_tissue_model,'uchar');
fclose(fid);

%% define detector position

col = (32:66)';
detector_position = zeros(48*size(col,1),3);
for loop = 1:size(col);
   c = col(loop);
   for slice = 1:48
      detector_position((loop-1)*48+slice,2)=slice+13; % y starts from 13
      detector_position((loop-1)*48+slice,1)=c;
      detector_position((loop-
1)*48+slice,3)=max(find(tagged_tissue_model(c,slice+13,:)==0))+1;
   end
end
clear col slice loop s c i ;
save('detector_position','detector_position');
```

```
%% define source position


col = (32:66)';
source_position = zeros(48*size(col,1),3);
for loop = 1:size(col);
   c = col(loop);
   for slice = 1:48
      source_position((loop-1)*48+slice,2)=slice+13; % y starts from 13
      source_position((loop-1)*48+slice,1)=c;
      source_position((loop-
1)*48+slice,3)=max(find(tagged_tissue_model(c,slice+13,:)==0))+1;
   end
end
clear col slice loop s c i ;
save('source_position','source_position');


%% tissue optical properties for baseline


w_sulcus = [
   18.18 0.89 0.04 1.37%   #1 Skull
   0.09 0.89 0.001 1.37%   #2 CSF
   37.5 0.84 0.005 1.37%   #3 White
   9.09 0.89 0.06 1.37 %   #4 Scalp
   9.09 0.89 0.06 1.37 %   #5 Scalp
   9.09 0.89 0.06 1.37 %   #6 Scalp
```

9.09 0.89 0.06 1.37 % #7 Scalp

9.09 0.89 0.06 1.37 % #8 Scalp

9.09 0.89 0.06 1.37 % #9 Scalp

9.09 0.89 0.06 1.37 % #10 Scalp

9.09 0.89 0.06 1.37 % #11 Scalp

9.09 0.89 0.06 1.37 % #12 Scalp

9.09 0.89 0.06 1.37 % #13 Scalp

9.09 0.89 0.06 1.37 % #14 Scalp

9.09 0.89 0.06 1.37 % #15 Scalp

9.09 0.89 0.06 1.37 % #16 Scalp

9.09 0.89 0.06 1.37 % #17 Scalp

9.09 0.89 0.06 1.37 % #18 Scalp

9.09 0.89 0.06 1.37 % #19 Scalp

9.09 0.89 0.06 1.37 % #20 Scalp

9.09 0.89 0.06 1.37 % #21 Scalp

9.09 0.89 0.06 1.37 % #22 Scalp

9.09 0.89 0.06 1.37 % #23 Scalp

22.72 0.89 0.025 1.37 % #24 Grey

22.72 0.89 0.025 1.37 % #25 Grey

22.72 0.89 0.025 1.37 % #26 Grey

22.72 0.89 0.025 1.37 % #27 Grey

22.72 0.89 0.025 1.37 % #28 Grey

22.72 0.89 0.025 1.37 % #29 Grey

22.72 0.89 0.025 1.37 % #30 Grey

22.72 0.89 0.025 1.37 % #31 Grey

```
    22.72 0.89 0.025 1.37 %   #32 Grey
    22.72 0.89 0.025 1.37 %   #33 Grey
    22.72 0.89 0.025 1.37 %   #34 Grey
    22.72 0.89 0.025 1.37 %   #35 Grey
    22.72 0.89 0.025 1.37 %   #36 Grey - Primary Motor
    22.72 0.89 0.025 1.37%   #37 Grey - Primary Somatosensory
    ];
save('tissue_opt_prop','w_sulcus');

%% Baseline Noise - Tissue absorption properties for 5 min physiological noise
load('Delta_MuA_part_3.mat');
Delta_MuA = w_sulcus(:,3)*ones(1,3110);
Delta_MuA(4:23,:) = Delta_MuA(4:23,:) + unique_Anat_Mua_S;
Delta_MuA(24:35,:) = Delta_MuA(24:35,:) + unique_Anat_Mua_G;
Delta_MuA(36,:) = Delta_MuA(36,:) + unique_Anat_Mua_G(11,:);
Delta_MuA(37,:) = Delta_MuA(37,:) + unique_Anat_Mua_G(11,:);

figure(1); subplot(2,1,2); plot(1:3110,Delta_MuA(36,:));

figure(2); plot(1:3110,Delta_MuA(24:37,:));
title('Grey Voxels Delta MuA activation signal'); xlabel('time samples');
ylabel('MuA (1/mm)');

figure(3); plot(1:3110,Delta_MuA(4:23,:));
```

```
        title('Scalp Voxels Delta MuA activation signal'); xlabel('time samples');
ylabel('MuA (1/mm)');

        file_nm = cat(2,'Baseline_Delta_MuA');
        save(file_nm, 'Delta_MuA');
```

MONTE CARLO SIMULATION

```
clc; clear all;

load('tissue_opt_prop.mat');
tissue_opt_prop = w_sulcus;
media_count = size(tissue_opt_prop,1);

load('source_position.mat');
src_count = size(source_position,1);

load('detector_position.mat');
det_count = size(detector_position,1);
det_radius = 0.5;
det_set = 1;
phot_set = 100;

random_seed = ones(phot_set,1);
for c=1:phot_set
   random_seed(c,1)=randi([2000 10000]);
end

src =[91];
clear c;
tic
```

```
for s = 1:4
    sim=src(s);
    x=source_position(sim,1);
    y=source_position(sim,2);
    z=source_position(sim,3);

    tic
    for loop = 1:phot_set
        iter = det_set;

        filenm = sprintf('tagged_tissue_%d_%d_%d_%d.inp',sim,x,y,z);
        fid = fopen(filenm,'wt');

        fprintf(fid,'1000000                    # No. of photons\n');
        fprintf(fid,'%d                    # RNG seed, negative to
generate\n',random_seed(loop));
        fprintf(fid,'%d %d %d             # x,z,y Position (mm) of the source\n',x,y,z);
        fprintf(fid,'0.0 0.0 1.0          # ux,uz,uy Direction cosin of the source\n');
        fprintf(fid,'0.e+00 12.e-09 12.e-09          # sample times and time intervel
(sec)\n');
        fprintf(fid,'tagged_tissue_model.bin   # volume filename, ASCII/Binary\n');
        fprintf(fid,'1 66 1 66                  # x: voxel size (isotropic only), dim,
start/end indices\n');
        fprintf(fid,'1 70 1 70                # y: voxel size, dim, start/end indices\n');
        fprintf(fid,'1 70 1 70                  # z: voxel size, dim, start/end indices\n');
```

```
fprintf(fid,'%d                    # No. of media\n', media_count);

for i = 1:media_count
   fprintf(fid,'%0.3f %0.2f %0.3f %0.2f     # scat(1/mm), g, mua (1/mm),
n\n',tissue_opt_prop(i,1),tissue_opt_prop(i,2),tissue_opt_prop(i,3),tissue_opt_prop(i,4));
end

fprintf(fid,'%d %0.2f       # detector number and radius (in grid
unit)\n',det_count, det_radius);
for j = 1+((iter-1)*det_count):det_count+((iter-1)*det_count)
   fprintf(fid,'%d %d %d   # detector position (in grid
unit)\n',detector_position(j,1),detector_position(j,2),detector_position(j,3));
end

% execute the mcx_det in cmd prompt
fclose(fid);
change_dir_cmd = 'cd
C:\Users\Bane\Documents\MATLAB\mcx\mcx\bin\Thesis\BaselineNoiseModel\Simulation
';
system(change_dir_cmd);
output_file_id = sprintf('tagged_tissue_%d_%d_%d_%d_%d',sim,loop,x,y,z);
exe_mcx_cmd = sprintf('mcx_det_128_1700 -A -n 1e6 -r 65 -f
tagged_tissue_%d_%d_%d_%d.inp -s %s -d 1 -G 1 -S 0 -E
%i',sim,x,y,z,output_file_id,random_seed(loop)); % saving flux field
system(exe_mcx_cmd);
```

```
    end
    toc
end
toc
```

```
clear source x y z media_count a detector_position source_position det_count
src_count det_radius phot_set det_set seed filenm fid change_dir_cmd output_file_id
exe_mcx_cmd ans sim iter loop i j;
```

POST PROCESSING SIMULATION OUTPUTS

```
% Load output from MCX & other inputs
clc;
clear all;

load('tissue_opt_prop.mat');
tissue_opt_prop = w_sulcus;
media_count = size(tissue_opt_prop,1);

load('source_position.mat');
src_count = size(source_position,1);

load('detector_position.mat');
det_count = size(detector_position,1);

det_set = 1;
phot_set = 100;
append_end = 0;

Ndet = det_set*det_count;

tic
src = [85;111;1151;1177];
sum_length = zeros(Ndet,media_count);
```

```
for i = 1:4
    sim = src(i);

    file_nm = cat(2,'Baseline_Delta_MuA');
    load(file_nm);

    x=source_position(sim,1);
    y=source_position(sim,2);
    z=source_position(sim,3);

    photon_count = zeros(Ndet,1);
    detfile = zeros(Ndet,7+size(Delta_MuA,2));
    detfile(:,1:3) = detector_position(:,1:3);
    detfile(:,5) = 1:Ndet;
    detfile(:,6) = 0;
    detfile(:,7) = getdistance([x,y,z],detfile(:,1:3));
    for loop = 1:phot_set
        load_data =
loadmch(sprintf('tagged_tissue_%d_%d_%d_%d_%d.mch',sim,loop,x,y,z));

        for d=1:Ndet
            scatter_length(:,1:media_count) =
load_data(find(detfile(d,5)==load_data(:,1)),3:(2+media_count));
            photon_count(d,1) = photon_count(d,:)+size(scatter_length,1);
            ex = exp(-(scatter_length*tissue_opt_prop(:,3)));
```

```
                ref = sum(ex);
                detfile(d,6) = detfile(d,6)+ref;

                ex2 = exp(-(scatter_length*Delta_MuA));
                ref2 = sum(ex2,1);
                detfile(d,8:3117) = detfile(d,8:3117)+ref2; % 6 baseline 7 distance 8
through 3117 is time series reflectance

                sum_length(d,:) = sum_length(d,:)+sum(scatter_length,1);
                clear scatter_length ex ref ex2 ref2;
            end
            clear load_data;
        end
        toc

        filenm =
cat(2,'baseline_MCX_postproc_brain','_N',num2str(sim),'_X',num2str(x),'_Y',num2str(y),'_
Z',num2str(z),'.mat');
        save(filenm,'detfile','sum_length','photon_count')
    end

    tic

    % for i = 1:4
    %     sim = src(i);
```

```
%
%    file_nm = cat(2,'Activation_Delta_MuA_16s');
%    load(file_nm);
%
%    x=source_position(sim,1);
%    y=source_position(sim,2);
%    z=source_position(sim,3);
%
%    photon_count = zeros(Ndet,1);
%    detfile = zeros(Ndet,7+size(Delta_MuA,2));
%    detfile(:,1:3) = detector_position(:,1:3);
%    detfile(:,5) = 1:Ndet;
%    detfile(:,6) = 0;
%    detfile(:,7) = getdistance([x,y,z],detfile(:,1:3));
%    for loop = 1:phot_set
%        load_data =
loadmch(sprintf('tagged_tissue_%d_%d_%d_%d_%d.mch',sim,loop,x,y,z));
%
%        for d=1:Ndet
%            scatter_length(:,1:media_count) =
load_data(find(detfile(d,5)==load_data(:,1)),3:(2+media_count));
%            photon_count(d,1) = photon_count(d,:)+size(scatter_length,1);
%            ex = exp(-(scatter_length*tissue_opt_prop(:,3)));
%            ref = sum(ex);
%            detfile(d,6) = detfile(d,6)+ref;
```

```
%
%               ex2 = exp(-(scatter_length*Delta_MuA));
%               ref2 = sum(ex2,1);
%               detfile(d,8:3117) = detfile(d,8:3117)+ref2; % 6 baseline 7 distance 8
through 3117 is time series reflectance
%
%               sum_length(d,:) = sum_length(d,:)+sum(scatter_length,1);
%               clear scatter_length ex ref ex2 ref2;
%       end
%       clear load_data;
%   end
%   toc
%
%   filenm =
cat(2,'activation_16s_MCX_postproc_brain','_N',num2str(sim),'_X',num2str(x),'_Y',num2s
tr(y),'_Z',num2str(z),'.mat');
%   save(filenm,'detfile','sum_length','photon_count')
% end
% clear all
```

ANALYSING RESULTS

```
clc;
clear;
close all;

%% Part - 1 %%
load('source_position.mat');
src_count=size(source_position,1);

load('tagged_tissue_model_new.mat');
load('Baseline_Delta_MuA.mat');

sim = 37;                   % source# for analysis
x = source_position(sim,1); y = source_position(sim,2); z =
source_position(sim,3);

load(sprintf('baseline_MCX_postproc_brain_N%d_X%d_Y%d_Z%d.mat',sim,x,y,
z));

%% Photon count Profile in 2-D

photon_escape =
zeros(size(tagged_tissue_model,1),size(tagged_tissue_model,2));
for i = 1:size(detfile,1)
```

```
    photon_escape(detfile(i,1),detfile(i,2)) = photon_count(i,1);
end
clear photon_count i;

plane_1 = zeros(size(tagged_tissue_model,1),size(tagged_tissue_model,2));
plane_2 = tagged_tissue_model(:,:,19); % transverse slice at depth ~ 11 mm

%% Identify detectors in-line with source and sulcus @ 1cm and beyond

a = find(abs(detfile(:,2)-source_position(sim,2))==10);
det_row = a(find(detfile(a,1)==source_position(sim,1)),1);  % det_row is detectors
@1cm
del_ref_s = -log(detfile(det_row(1,1),108:3017)./detfile(det_row(1,1),6)); % -
loge(exp(MuA+Del_MuA)L/exp(MuA)L) % Reference noise

b = find(abs(detfile(:,7))>=11);
c = find(abs(detfile(:,2)-source_position(sim,2))>=10); % filt_row are detectors
beyond @1cm
filt_row = b;%(find(detfile(c,1)==source_position(sim,1)),1);
% filt_row = b;

%% Adaptive Filtering scalp noise from grey

close all;
mu = 0.000005;
```

```
L = 300;
noise = del_ref_s';
noise = (noise - mean(noise))/std(noise);
for i = 1:length(b)
   sig = detfile(b(i),108:3017)';
   sigf = (sig - mean(sig))/std(sig);
   [o(:,i), e(:,i)] = LMSFilter(mu, L, sigf, noise); %filter all detectors beyond 1cm
   o(:,i) = (o(:,i) * std(sig))+ mean(sig);
end
filt_detfile = detfile;
filt_detfile(b,108:3017)=o';

loc = 218; %find(b == 81);

%% Plot 1-D reflectance post filtering

% Calculate 2-D reflectance post filtering
filt_plane =
zeros(size(tagged_tissue_model,1),size(tagged_tissue_model,2),2910);
for row = 1:size(filt_detfile,1)
   filt_plane(filt_detfile(row,1),filt_detfile(row,2),:)=filt_detfile(row,108:3017);
end
clear row;

%% Statistical Check for effect of Physiological noise
```

```
        err_pct =
std(detfile(:,108:3017),0,2)./(mean(detfile(:,108:3017),2)*sqrt(length(108:3017)))*100;
        err_pct_filt =
std(filt_detfile(:,108:3017),0,2)./(mean(filt_detfile(:,108:3017),2)*sqrt(length(108:3017)))*1
00;
        clear detfile;

        % Plot 2-d std deviation on surface - before and after filtering
        plane_4 = zeros(size(tagged_tissue_model,1),size(tagged_tissue_model,2));
        plane_5 = zeros(size(tagged_tissue_model,1),size(tagged_tissue_model,2));
        for row = 1:size(filt_detfile,1)
           plane_4(filt_detfile(row,1),filt_detfile(row,2))=err_pct(row,1);
           plane_5(filt_detfile(row,1),filt_detfile(row,2))=err_pct_filt(row,1);
        end

        %% calculate 2-D Delta O.D at detectors

        baseline = filt_plane;
        save(sprintf('baseline_N%d_X%d_Y%d_Z%d.mat',sim,x,y,z),'baseline');
        %% Part -2 %%
        % load('Baseline_Delta_MuA.mat');       % for activation signal reference only
        %
load(sprintf('baseline_MCX_postproc_brain_N%d_X%d_Y%d_Z%d.mat',sim,x,y,z));

        load('Activation_Delta_MuA_16s.mat');       % for activation signal reference only
```

```
load(sprintf('activation_16s_MCX_postproc_brain_N%d_X%d_Y%d_Z%d.mat',si
m,x,y,z));

%% Calculate Photon count in 2-D

photon_escape =
zeros(size(tagged_tissue_model,1),size(tagged_tissue_model,2));
for i = 1:size(detfile,1)
   photon_escape(detfile(i,1),detfile(i,2)) = photon_count(i,1);
end
clear photon_count i;

% %% Plot 1-D reflectance with noise
% detfile_subset = detfile(find(detfile(:,1)==source_position(sim,1)),:);
% figure(10+sim); subplot(2,1,1); plot(detfile_subset(:,7
),log10(detfile_subset(:,6)),'*b');
% hold on;
plot(detfile_subset(:,7),log10(detfile_subset(:,108:3017)),'*r','MarkerSize',2);
% ylabel ('Reflectance unit?'); xlabel('source-detector seperation (mm)');
% title(sprintf('1-D Reflectance comparison for effect of physiological noise |
Source#%d X%d Y%d Z%d',sim,x,y,z));
% legend('no-noise baseline','physiologiocal noise baseline'); xlim([10,35]);

% %% Plot 2-D reflectance and overlap on anatomy
% plane_1 = zeros(size(tagged_tissue_model,1),size(tagged_tissue_model,2));
```

```
plane_2 = tagged_tissue_model(:,:,19);
% plane_2_overlap = plane_2;
% for i = 1:size(detfile,1)
%     plane_1(detfile(i,1),detfile(i,2)) = log10(detfile(i,6));
%     plane_2_overlap(detfile(i,1),detfile(i,2)) =
plane_2(detfile(i,1),detfile(i,2))+(90*log10(detfile(i,6)));
% end
% figure(20+sim); imagesc(plane_1);
% title(sprintf('2-D reflectance at detectors on surface | Source#%d X%d Y%d
Z%d',sim,x,y,z));
% figure(30+sim); imagesc(plane_2_overlap);
% title(sprintf('2-D reflectance overlapped on Transverse slice at depth ~ 9mm |
Source#%d X%d Y%d Z%d',sim,x,y,z));
figure(40+sim); imagesc(plane_2); title('Transverse slice at depth ~ 9mm');
% clear  plane_2 plane_2_overlap i;

%% Identify detectors in-line with source and sulcus @ 1cm and beyond

a = find(abs(detfile(:,2)-source_position(sim,2))==10);
det_row = a(find(detfile(a,1)==source_position(sim,1)),1);  % det_row is detectors
@1cm
del_ref_s = -log(detfile(det_row(1,1),108:3017)./detfile(det_row(1,1),6)); % -
loge(exp(MuA+Del_MuA)L/exp(MuA)L) % Reference noise

b = find(abs(detfile(:,7))>=11);
```

```
        c = find(abs(detfile(:,2)-source_position(sim,2))>=10); % filt_row are detectors
beyond @1cm
        filt_row = b;%(find(detfile(c,1)==source_position(sim,1)),1);
        % filt_row = b;

        %% Adaptive Filtering scalp noise from grey

        close all;

        noise = del_ref_s';
        noise = (noise - mean(noise))/std(noise);
        for i = 1:length(b)
           sig = detfile(b(i),108:3017)';
           sigf = (sig - mean(sig))/std(sig);
           [o(:,i), e(:,i)] = LMSFilter(mu, L, sigf, noise); %filter all detectors beyond 1cm
           o(:,i) = (o(:,i) * std(sig))+ mean(sig);
        end
        filt_detfile = detfile;
        filt_detfile(b,108:3017)=o';

        loc = 218; %find(b == 81);

        % % Plot effects of filtering
        % figure(50+sim); subplot(2,2,1);
        % plot(101:3010,del_ref_s'); title('noise');
```

```
        % subplot(2,2,2); hold on;
        % plot(101:3010,detfile(loc,108:3017)','y'); title('original signal');
        % plot(101:3010,filt_detfile(loc,108:3017),'r'); title('filtered output'); hold off;
        % subplot(2,2,3);
        % plot(101:3010,e(:,loc)); title('error');
        % subplot(2,2,4);
        % plot(101:3010,Delta_MuA(36,101:3010),'r'); title('activation MuA for ref');
        figure(55+sim); subplot(2,1,1); plot(101:3010,Delta_MuA(36,101:3010),'r');
title('activation MuA for ref');grid on;
        %% Plot 1-D reflectance post filtering

        % figure(10+sim); subplot(2,1,2); hold on;
        % plot(detfile(filt_row,7),log10(detfile(filt_row,6)),'*b');
        %
plot(filt_detfile(filt_row,7),log10(filt_detfile(filt_row,108:3017)),'*r','MarkerSize',2);grid on;
        % ylabel ('Reflectance unit?'); xlabel('source-detector seperation (mm)');
        % title(sprintf('1-D Filtered Reflectance | Source#%d X%d Y%d Z%d',sim,x,y,z));
        % legend('Filtered baseline','Filtered physiological noise'); xlim([10,35]);
        % clear a b mu L e;

        % Calculate 2-D reflectance post filtering
        filt_plane =
zeros(size(tagged_tissue_model,1),size(tagged_tissue_model,2),2910);
        for row = 1:size(filt_detfile,1)
            filt_plane(filt_detfile(row,1),filt_detfile(row,2),:)=filt_detfile(row,108:3017);
```

```
        end
        clear row;

        %% Statistical Check for effect of Physiological noise
        err_pct =
std(detfile(:,108:3017),0,2)./(mean(detfile(:,108:3017),2)*sqrt(length(108:3017)))*100;
        err_pct_filt =
std(filt_detfile(:,108:3017),0,2)./(mean(filt_detfile(:,108:3017),2)*sqrt(length(108:3017)))*1
00;
        clear detfile;

        % % Plot 1-D error along in-line detectors
        % figure(100+sim); plot(filt_detfile(filt_row,7),err_pct(filt_row),'*b');
        % hold on; plot(filt_detfile(filt_row,7),err_pct_filt(filt_row),'*g');
        % ylabel('Error %'); xlabel('Source-detector seperation distance (mm)');
        % title(sprintf('Pct Error before and after filtering | Source#%d X%d Y%d
Z%d',sim,x,y,z));
        % legend('before filtering','after filtering');

        % Plot 2-d std deviation on surface - before and after filtering
        plane_4 = zeros(size(tagged_tissue_model,1),size(tagged_tissue_model,2));
        plane_5 = zeros(size(tagged_tissue_model,1),size(tagged_tissue_model,2));
        for row = 1:size(filt_detfile,1)
            plane_4(filt_detfile(row,1),filt_detfile(row,2))=err_pct(row,1);
            plane_5(filt_detfile(row,1),filt_detfile(row,2))=err_pct_filt(row,1);
```

```
        end
        % % figure(110+sim); imagesc(plane_4); title(sprintf('Standard deviation before
filtering | Source#%d X%d Y%d Z%d',sim,x,y,z)); colorbar;
        % % figure(120+sim); imagesc(plane_5); title(sprintf('Standard deviation after
filtering | Source#%d X%d Y%d Z%d',sim,x,y,z)); colorbar;

        %% calculate 2-D Delta O.D at detectors

        load(sprintf('baseline_N%d_X%d_Y%d_Z%d.mat',sim,x,y,z));
        mean_baseline = mean(baseline(30:58,19:51,:),3);
        DOD10 = ones(size(filt_plane));

        for t = 1:2910
           DOD10(30:58,19:51,t) = (filt_plane(30:58,19:51,t))./mean_baseline;
        end

        keep_DOD_1 = -log10(DOD10);
        keep_dod_1 = zeros(1107,size(keep_DOD_1,3));
        i=1;
        for row = min(filt_detfile(:,1)):max(filt_detfile(:,1))
           for col = min(filt_detfile(:,2)):max(filt_detfile(:,2))
              keep_dod_1(i,:) = keep_DOD_1(row,col,:);
              i = i+1;
           end
        end
```

```
figure(55+sim)
subplot(2,1,2); hold on; plot(101:3010,keep_dod_1(306,:),'b');
grid on;
hold off;
filenm = sprintf('0s_dod_N%d_X%d_Y%d_Z%d.mat',sim,x,y,z);
save(filenm,'keep_dod_1','keep_DOD_1');

[keep_DOD_3,keep_DOD_7,keep_DOD_11,keep_DOD_13,keep_DOD_17,keep
_dod_3,keep_dod_7,keep_dod_11,keep_dod_13,keep_dod_17] =
moving_summation(filt_detfile,filt_plane,keep_DOD_1,baseline);
view3dgui(keep_DOD_13);
clear row t DOD10 baseline;


%% 1-D plot of Delta O.D for groups of 3, 5, 7, 9 and 11 detectors

loc = 306;
figure(130+sim); subplot(2,1,1);
plot(1:2910,Delta_MuA(36,101:3010));title('Primary Motor cortex activation
signal'); xlabel('Times Samples'); ylabel('MuA (1/mm)');

% for i = 1:35
subplot(2,1,2); hold on;
plot(1:2910,keep_dod_1(loc,1:2910),'b');
plot(1:2910,keep_dod_3(loc,1:2910),'g');
```

```
plot(1:2910,keep_dod_7(loc,1:2910),'c');
%      plot(1:2910,keep_dod_9(loc,1:2910),'m');
plot(1:2910,keep_dod_11(loc,1:2910),'r');
plot(1:2910,keep_dod_13(loc,1:2910),'k');
plot(1:2910,keep_dod_17(loc,1:2910),'m');
legend('1 mm','3 mm','7 mm','11 mm','13 mm','17mm');
% end

%% Identify centrally summed detector that detects most

a=Delta_MuA(36,101:3010);              % Delta MuA activation signal as ref
Gref(:,1) = gpuArray(a);
for row = 1:size(keep_DOD_3,1)
   for col = 1:size(keep_DOD_3,2)
      q(1,:) = keep_DOD_3(row,col,1:2910); % moving summed delta O.D at
detectors for comparison
      Gref(:,2) = gpuArray(q);
      Gtest = xcorr(Gref(:,1),Gref(:,2),'coeff');
      [argvalue] = max(gather(Gtest));
      DOD_3_tag(row,col) = argvalue;
   end
end

% 2-D plot of correlated detectors
figure(140+sim); imagesc(DOD_3_tag); colorbar;
```

```
title(sprintf('Detectors detecting from primary-motor activation | Source#%d X%d Y%d Z%d',sim,x,y,z));

clear a q Gref Gtest argvalue argmax row col;

%% See delta O.D change for different sum sizes

sel_x = 47; sel_y = 47;      % detector co-ords for analysis
fs = 10.35;
pretime = 5;
a=Delta_MuA(36,:);
start = 25;

box(1,:) = keep_DOD_3(sel_x,sel_y,:);       % orig
box(2,:) = keep_DOD_3(sel_x,sel_y,:);       % 3*3 box
box(3,:) = keep_DOD_7(sel_x,sel_y,:);       % 5*5 box
box(4,:) = keep_DOD_7(sel_x,sel_y,:);       % 7*7 box
box(5,:) = keep_DOD_11(sel_x,sel_y,:);       % 9*9 box
box(6,:) = keep_DOD_11(sel_x,sel_y,:);      % 11*11 box
box(7,:) = keep_DOD_13(sel_x,sel_y,:);      % 13*13 box
box(8,:) = keep_DOD_17(sel_x,sel_y,:);      % 17*17 box
box(9,:) = keep_DOD_17(sel_x,sel_y,:);      % 21*21 box

stim = [start start+32 start+64 start+96 start+128 start+160 start+192]-10;%-10 to account for clipping
```

```
        mean_base_1 = zeros(size(box,1),1);
        mean_base_2 = zeros(size(box,1),1);
        mean_base_3 = zeros(size(box,1),1);
        mean_base_4 = zeros(size(box,1),1);
        mean_base_5 = zeros(size(box,1),1);
        mean_base_6 = zeros(size(box,1),1);
        mean_base_7 = zeros(size(box,1),1);

        for detgrp = 1:size(box,1)
            mean_base_1(detgrp,:) = mean(box(detgrp,round((stim(1) -
pretime)*fs):round(stim(1)*fs)),2);
            mean_base_2(detgrp,:) = mean(box(detgrp,round((stim(2) -
pretime)*fs):round(stim(2)*fs)),2);
            mean_base_3(detgrp,:) = mean(box(detgrp,round((stim(3) -
pretime)*fs):round(stim(3)*fs)),2);
            mean_base_4(detgrp,:) = mean(box(detgrp,round((stim(4) -
pretime)*fs):round(stim(4)*fs)),2);
            mean_base_5(detgrp,:) = mean(box(detgrp,round((stim(5) -
pretime)*fs):round(stim(5)*fs)),2);
            mean_base_6(detgrp,:) = mean(box(detgrp,round((stim(6) -
pretime)*fs):round(stim(6)*fs)),2);
            mean_base_7(detgrp,:) = mean(box(detgrp,round((stim(7) -
pretime)*fs):round(stim(7)*fs)),2);
```

```
            block_1(detgrp,:) = box(detgrp,round((stim(1) - pretime)*fs):round(stim(2)*fs))-
mean_base_1(detgrp,:);
            block_2(detgrp,:) = box(detgrp,round((stim(2) - pretime)*fs):round(stim(3)*fs))-
mean_base_2(detgrp,:);
            block_3(detgrp,:) = box(detgrp,round((stim(3) - pretime)*fs):round(stim(4)*fs))-
mean_base_3(detgrp,:);
            block_4(detgrp,:) = box(detgrp,round((stim(4) - pretime)*fs):round(stim(5)*fs))-
mean_base_4(detgrp,:);
            block_5(detgrp,:) = box(detgrp,round((stim(5) - pretime)*fs):round(stim(6)*fs))-
mean_base_5(detgrp,:);
            block_6(detgrp,:) = box(detgrp,round((stim(6) - pretime)*fs):round(stim(7)*fs))-
mean_base_6(detgrp,:);
            block_7(detgrp,:) = box(detgrp,round((stim(7) - pretime)*fs):round((stim(7) +
32)*fs)) - mean_base_7(detgrp,:);
        end

        % Plot individual blocks
        figure(200+sim); hold on;
        plot(block_1(1,:),'k'); plot(block_2(1,:),'b'); plot(block_3(1,:),'c');
        plot(block_4(1,:),'m'); plot(block_5(1,:),'r'); plot(block_6(1,:),'y');
        plot(block_7(1,:),'g'); hold off; title ('Individual epoch blocks');
        legend('block1','block2','block3','block4','block5','block6','block7');

        % Average of tapping blocks
        avg_block_1 = block_3(:,1:383);
```

```
        avg_block_2 = (block_3(:,1:383)+block_2(:,1:383))/2;
        avg_block_3 = (block_3(:,1:383)+block_2(:,1:383)+block_1(:,1:383))/3;
        avg_block_4 =
(block_3(:,1:383)+block_2(:,1:383)+block_1(:,1:383)+block_6(:,1:383))/4; % originally
block_5
        avg_block_5 =
(block_3(:,1:383)+block_2(:,1:383)+block_1(:,1:383)+block_6(:,1:383)+block_7(:,1:383))/
5;
        avg_block_6 =
(block_3(:,1:383)+block_2(:,1:383)+block_1(:,1:383)+block_6(:,1:383)+block_7(:,1:383)+
block_5(:,1:383))/6;
        avg_block_7 =
(block_3(:,1:383)+block_2(:,1:383)+block_1(:,1:383)+block_6(:,1:383)+block_7(:,1:383)+
block_5(:,1:383)+block_4(:,1:383))/7;

        % Plot average blocks
        figure(210+sim); hold on;
        plot(avg_block_1(3,:),'k'); plot(avg_block_2(3,:),'b'); plot(avg_block_3(3,:),'c');
        plot(avg_block_4(3,:),'m'); plot(avg_block_5(3,:),'r'); plot(avg_block_6(3,:),'y');
        plot(avg_block_7(3,:),'g');  title ('Average epoch blocks');
        legend('avgblock1','avgblock2','avgblock3','avgblock4','avgblock5','avgblock6','av
gblock7');
        hold off;

        % finding location of max of averaged pulse (seconds)
```

```
            max_range = round(15.0*fs:16.0*fs);
            length_base = round(6*fs);
            length_max = round(0.5*fs);
            sample_max = zeros(size(box,1),7);
            max_start = zeros(size(box,1),7);
            max_end = zeros(size(box,1),7);

            for detgrp = 1:size(box,1)
                sample_max(detgrp,1) =
find(avg_block_1(detgrp,:)==max(avg_block_1(detgrp,max_range),[],2));%round(14*fs);
% max value co-ordinate of activation pulses for reference
                sample_max(detgrp,2) =
find(avg_block_2(detgrp,:)==max(avg_block_2(detgrp,max_range),[],2));
                sample_max(detgrp,3) =
find(avg_block_3(detgrp,:)==max(avg_block_3(detgrp,max_range),[],2));
                sample_max(detgrp,4) =
find(avg_block_4(detgrp,:)==max(avg_block_4(detgrp,max_range),[],2));
                sample_max(detgrp,5) =
find(avg_block_5(detgrp,:)==max(avg_block_5(detgrp,max_range),[],2));
                sample_max(detgrp,6) =
find(avg_block_6(detgrp,:)==max(avg_block_6(detgrp,max_range),[],2));
                sample_max(detgrp,7) =
find(avg_block_7(detgrp,:)==max(avg_block_7(detgrp,max_range),[],2));
                max_start(detgrp,:)= sample_max(detgrp,:) - length_max*ones(1,7);
                max_end(detgrp,:) = sample_max(detgrp,:) + length_max*ones(1,7);
```

```
end

% Calculate mean of max, std dev of max and std dev of baseline
mean_max = zeros(size(box,1),7);
std_max = zeros(size(box,1),7);
std_base = zeros(size(box,1),7);

for detgrp = 1:size(box,1)
   mean_max(detgrp,1) =
mean(avg_block_1(detgrp,max_start(detgrp,1):max_end(detgrp,1)),2);
   std_max(detgrp,1) =
std(avg_block_1(detgrp,max_start(detgrp,1):max_end(detgrp,1)),0,2);
   std_base(detgrp,1) = std(avg_block_1(detgrp,1:length_base),0,2);

   mean_max(detgrp,2) =
mean(avg_block_2(detgrp,max_start(detgrp,1):max_end(detgrp,1)),2);
   std_max(detgrp,2) =
std(avg_block_2(detgrp,max_start(detgrp,1):max_end(detgrp,1)),0,2);
   std_base(detgrp,2) = std(avg_block_2(detgrp,1:length_base),0,2);

   mean_max(detgrp,3) =
mean(avg_block_3(detgrp,max_start(detgrp,1):max_end(detgrp,1)),2);
   std_max(detgrp,3) =
std(avg_block_3(detgrp,max_start(detgrp,1):max_end(detgrp,1)),0,2);
   std_base(detgrp,3) = std(avg_block_3(detgrp,1:length_base),0,2);
```

```
            mean_max(detgrp,4) =
mean(avg_block_4(detgrp,max_start(detgrp,1):max_end(detgrp,1)),2);
            std_max(detgrp,4) =
std(avg_block_4(detgrp,max_start(detgrp,1):max_end(detgrp,1)),0,2);
            std_base(detgrp,4) = std(avg_block_4(detgrp,1:length_base),0,2);

            mean_max(detgrp,5) =
mean(avg_block_5(detgrp,max_start(detgrp,1):max_end(detgrp,1)),2);
            std_max(detgrp,5) =
std(avg_block_5(detgrp,max_start(detgrp,1):max_end(detgrp,1)),0,2);
            std_base(detgrp,5) = std(avg_block_5(detgrp,1:length_base),0,2);

            mean_max(detgrp,6) =
mean(avg_block_6(detgrp,max_start(detgrp,1):max_end(detgrp,1)),2);
            std_max(detgrp,6) =
std(avg_block_6(detgrp,max_start(detgrp,1):max_end(detgrp,1)),0,2);
            std_base(detgrp,6) = std(avg_block_6(detgrp,1:length_base),0,2);

            mean_max(detgrp,7) =
mean(avg_block_7(detgrp,max_start(detgrp,1):max_end(detgrp,1)),2);
            std_max(detgrp,7) =
std(avg_block_7(detgrp,max_start(detgrp,1):max_end(detgrp,1)),0,2);
            std_base(detgrp,7) = std(avg_block_7(detgrp,1:length_base),0,2);
        end
```

```
snr = mean_max/0.0013; % averaged blocks

% 1-D SNR Plot for averaged tapping blocks
figure(180+sim); plot(1:5,snr(:,1:5)'); xlim([0 7]);
title('Signal to noise ratio per activation block averages for grouped detectors');
legend('1mm','3mm','5mm','7mm','9mm','11mm','13mm','17mm','21mm');
xlabel('Average Epochs'); ylabel('SNR');

% Calculate mean of max std of max and std of baseline
ind_mean_max = zeros(size(box,1),7);
ind_std_max = zeros(size(box,1),7);
ind_std_base = zeros(size(box,1),7);

for detgrp = 1:size(box,1)
   ind_mean_max(detgrp,1) =
mean(block_1(detgrp,max_start(detgrp,1):max_end(detgrp,1)),2);
   ind_std_max(detgrp,1) =
std(block_1(detgrp,max_start(detgrp,1):max_end(detgrp,1)),0,2);
   ind_std_base(detgrp,1) = std(block_1(detgrp,1:length_base),0,2);

   ind_mean_max(detgrp,2) =
mean(block_2(detgrp,max_start(detgrp,1):max_end(detgrp,1)),2);
   ind_std_max(detgrp,2) =
std(block_2(detgrp,max_start(detgrp,1):max_end(detgrp,1)),0,2);
```

```
            ind_std_base(detgrp,2) = std(block_2(detgrp,1:length_base),0,2);


            ind_mean_max(detgrp,3) =
mean(block_3(detgrp,max_start(detgrp,1):max_end(detgrp,1)),2);
            ind_std_max(detgrp,3) =
std(block_3(detgrp,max_start(detgrp,1):max_end(detgrp,1)),0,2);
            ind_std_base(detgrp,3) = std(block_3(detgrp,1:length_base),0,2);


            ind_mean_max(detgrp,4) =
mean(block_4(detgrp,max_start(detgrp,1):max_end(detgrp,1)),2);
            ind_std_max(detgrp,4) =
std(block_4(detgrp,max_start(detgrp,1):max_end(detgrp,1)),0,2);
            ind_std_base(detgrp,4) = std(block_4(detgrp,1:length_base),0,2);


            ind_mean_max(detgrp,5) =
mean(block_5(detgrp,max_start(detgrp,1):max_end(detgrp,1)),2);
            ind_std_max(detgrp,5) =
std(block_5(detgrp,max_start(detgrp,1):max_end(detgrp,1)),0,2);
            ind_std_base(detgrp,5) = std(block_5(detgrp,1:length_base),0,2);


            ind_mean_max(detgrp,6) =
mean(block_6(detgrp,max_start(detgrp,1):max_end(detgrp,1)),2);
            ind_std_max(detgrp,6) =
std(block_6(detgrp,max_start(detgrp,1):max_end(detgrp,1)),0,2);
            ind_std_base(detgrp,6) = std(block_6(detgrp,1:length_base),0,2);
```

```
            ind_mean_max(detgrp,7) =
mean(block_7(detgrp,max_start(detgrp,1):max_end(detgrp,1)),2);
            ind_std_max(detgrp,7) =
std(block_7(detgrp,max_start(detgrp,1):max_end(detgrp,1)),0,2);
            ind_std_base(detgrp,7) = std(block_7(detgrp,1:length_base),0,2);
        end

        ind_snr = ind_mean_max/0.0013; % averaged blocks

        % 1-D std deviation SNR Plot
        figure(190+sim);
        plot(1:7,ind_snr'); xlim([0 7]);
        title('Signal to noise ratio variations per activation block averages for grouped
detectors'); legend('1*1','3*3','5*5','7*7','9*9','11*11');
        xlabel('Individual Blocks'); ylabel('SNR');

        % Count escaped photons from grouped detectors
        sum_photon = sum(sum(photon_escape(35:46,13:53)));



        % %% figures
        % load('tissue_model.mat');
        % image = permute(tissue_model,[2 3 1]);
        % for x = 1:66
```

```
%     rev_image(x,:,:) = image(67-x,:,:);
% end
% figure(1); imagesc(rev_image(:,:,42)');title('Vertical slice through central sulcus'); xlabel('width (mm)');ylabel('depth (mm)');grid on;
% figure(2); plot(filt_detfile(filt_row,7),log10(filt_detfile(filt_row,6)),'*'); xlabel('Source-Detector Separation (mm)');ylabel('Reflectance (J/mm2)');
% figure(3); plot(10:35,plane_5(42,10:35),'*'); title('Standard deviation profile along series of detectors'); xlabel('source-detector seperation (mm)'); ylabel('standard deviation (%)');
% figure(4); plot(10:35,DOD_3_tag(42,10:35)*100,'*'); title('Cross-correlation profile with activation signal along series of detectors'); xlabel('source-detector seperation (mm)'); ylabel('correlation co-efficient(%)');
```

References

1. Zhang, Q., E.N. Brown, and G.E. Strangman, *Adaptive filtering for global interference cancellation and real-time recovery of evoked brain activity: a Monte Carlo simulation study.* Journal of Biomedical Optics, 2007. **12**(4): p. 044014-044014-12.
2. Zhang, Q., G.E. Strangman, and G. Ganis, *Adaptive filtering to reduce global interference in non-invasive NIRS measures of brain activation: How well and when does it work?* NeuroImage, 2009. **45**(3): p. 788-794.
3. Ferrari, M. and V. Quaresima, *A brief review on the history of human functional near-infrared spectroscopy (fNIRS) development and fields of application.* NeuroImage, 2012. **63**(2): p. 921-935.
4. Homan, R.W., J. Herman, and P. Purdy, *Cerebral location of international 10–20 system electrode placement.* Electroencephalography and Clinical Neurophysiology, 1987. **66**(4): p. 376-382.
5. Selb, J., et al., *Comparison of a layered slab and an atlas head model for Monte Carlo fitting of time-domain near-infrared spectroscopy data of the adult head.* Journal of Biomedical Optics, 2014. **19**(1): p. 016010-016010.
6. Soltysik, D.A., et al., *Comparison of hemodynamic response nonlinearity across primary cortical areas.* NeuroImage, 2004. **22**(3): p. 1117-1127.
7. Schweiger, M., A. Gibson, and S.R. Arridge, *Computational aspects of diffuse optical tomography.* Computing in Science & Engineering, 2003. **5**(6): p. 33-41.
8. Siegel, A., J.J. Marota, and D. Boas, *Design and evaluation of a continuous-wavediffuse optical tomography system.* Optics Express, 1999. **4**(8): p. 287-298.
9. Montcel, B., R. Chabrier, and P. Poulet, *Detection of cortical activation with time-resolved diffuse optical methods.* Applied Optics, 2005. **44**(10): p. 1942-1947.
10. Boas, D.A., A.M. Dale, and M.A. Franceschini, *Diffuse optical imaging of brain activation: approaches to optimizing image sensitivity, resolution, and accuracy.* NeuroImage, 2004. **23, Supplement 1**(0): p. S275-S288.
11. Saager, R.B. and A.J. Berger, *Direct characterization and removal of interfering absorption trends in two-layer turbid media.* Journal of the Optical Society of America A, 2005. **22**(9): p. 1874-1882.
12. Zhang, Y., et al., *Eigenvector-based spatial filtering for reduction of physiological interference in diffuse optical imaging.* Journal of Biomedical Optics, 2005. **10**(1): p. 011014-01101411.
13. Buxton, R.B., *The Elusive Initial Dip.* NeuroImage, 2001. **13**(6): p. 953-958.
14. Julien, C., *The enigma of Mayer waves: Facts and models.* Cardiovascular Research, 2006. **70**(1): p. 12-21.
15. Khan, B., et al. *Functional near infrared brain imaging with a brush-fiber optode to improve optical contact on subjects with dense hair*. 2011.
16. Bunce, S.C., et al., *Functional near-infrared spectroscopy.* Engineering in Medicine and Biology Magazine, IEEE, 2006. **25**(4): p. 54-62.
17. Ntziachristos, V., *Going deeper than microscopy: the optical imaging frontier in biology.* Nat Meth, 2010. **7**(8): p. 603-614.
18. Franceschini, M.A., et al., *Hemodynamic evoked response of the sensorimotor cortex measured noninvasively with near-infrared optical imaging.* Psychophysiology, 2003. **40**(4): p. 548-560.
19. Huppert, T.J., et al., *HomER: a review of time-series analysis methods for near-infrared spectroscopy of the brain.* Applied Optics, 2009. **48**(10): p. D280-D298.

20. Silverthorn, D.U., *Human Physiology: An Integrated Approach*. Vol. 6 edition. 2012: Pearson. 992 pages.
21. Khan, B., et al. *Identification of Abnormal Motor Cortex Activation Patterns in Children with Cerebral Palsy by Functional Near Infrared Spectroscopy*. in *Biomedical Optics and 3-D Imaging*. 2010. Miami, Florida: Optical Society of America.
22. Konecky, S.D., et al., *Imaging complex structures with diffuse light.* Optics Express, 2008. **16**(7): p. 5048-5060.
23. Khan, B., et al., *Improving optical contact for functional near infrared brain spectroscopy and imaging with brush optodes.* Biomedical Optics Express, 2012. **3**(5): p. 878-898.
24. Rolfe, P., *IN VIVO NEAR-INFRARED SPECTROSCOPY.* Annual Review of Biomedical Engineering, 2000. **2**(1): p. 715-754.
25. Tong, Y., L.M. Hocke, and B.d. Frederick, *Isolating the sources of widespread physiological fluctuations in functional near-infrared spectroscopy signals.* Journal of Biomedical Optics, 2011. **16**(10): p. 106005-106005-11.
26. Liemert, A. and A. Kienle, *Light diffusion in N-layered turbid media: frequency and time domains.* Journal of Biomedical Optics, 2010. **15**(2): p. 025002-025002-8.
27. Hoge, R.D., et al., *Linear coupling between cerebral blood flow and oxygen consumption in activated human cortex.* Proceedings of the National Academy of Sciences, 1999. **96**(16): p. 9403-9408.
28. Tian, F., S. Prajapati, and H. Liu. *A Location-Adaptive, Frequency-Specific Cancellation Algorithm to Improve Optical Brain Functional Imaging*. in *Biomedical Optics*. 2008. St. Petersburg, Florida: Optical Society of America.
29. LI, T., H. GONG, and Q. LUO, *MCVM: MONTE CARLO MODELING OF PHOTON MIGRATION IN VOXELIZED MEDIA.* Journal of Innovative Optical Health Sciences, 2010. **03**(02): p. 91-102.
30. Fukui, Y., Y. Ajichi, and E. Okada, *Monte Carlo prediction of near-infrared light propagation in realistic adult and neonatal head models.* Applied Optics, 2003. **42**(16): p. 2881-2887.
31. Fang, Q. and D.A. Boas, *Monte Carlo simulation of photon migration in 3D turbid media accelerated by graphics processing units.* Optics Express, 2009. **17**(22): p. 20178-20190.
32. Filosa, J.A. and V.M. Blanco, *Neurovascular coupling in the mammalian brain.* Experimental physiology, 2007. **92**(4): p. 641-646.
33. Strangman, G., D.A. Boas, and J.P. Sutton, *Non-invasive neuroimaging using near-infrared light.* Biological Psychiatry, 2002. **52**(7): p. 679-693.
34. Miller, K.L., et al., *Nonlinear temporal dynamics of the cerebral blood flow response.* Human Brain Mapping, 2001. **13**(1): p. 1-12.
35. Orihuela-Espina, F., et al., *Quality control and assurance in functional near infrared spectroscopy (fNIRS) experimentation.* Physics in Medicine and Biology, 2010. **55**(13): p. 3701.
36. A P Gibson1, J.C.H.a.S.R.A., *Recent advances in diffuse optical imaging.* Phys. Med. Biol., 2005. **50**(R1).
37. Zhang, Y., J.W. Sun, and P. Rolfe, *RLS adaptive filtering for physiological interference reduction in NIRS brain activity measurement: a Monte Carlo study.* Physiological Measurement, 2012. **33**(6): p. 925.
38. Hoge, R.D., et al., *Simultaneous recording of task-induced changes in blood oxygenation, volume, and flow using diffuse optical imaging and arterial spin-labeling MRI.* NeuroImage, 2005. **25**(3): p. 701-707.
39. Habermehl, C., et al., *Somatosensory activation of two fingers can be discriminated with ultrahigh-density diffuse optical tomography.* NeuroImage, 2012. **59**(4): p. 3201-3211.

40. Toyoda, H., et al., *Source of nonlinearity of the BOLD response revealed by simultaneous fMRI and NIRS.* NeuroImage, 2008. **39**(3): p. 997-1013.
41. Ashburner, J., et al., *SPM8 Manual*, U. Institute of Neurology, Editor. 2013: 12 Queen Square, London WC1N 3BG, UK.
42. Obrig, H., et al., *Spontaneous Low Frequency Oscillations of Cerebral Hemodynamics and Metabolism in Human Adults.* NeuroImage, 2000. **12**(6): p. 623-639.
43. Zhang, X. *Super-resolution method for arbitrary retrospective sampling in fluorescence tomography with raster scanning photodetectors*. 2013.
44. Okada, E., et al., *Theoretical and experimental investigation of near-infrared light propagation in a model of the adult head.* Applied Optics, 1997. **36**(1): p. 21-31.
45. Boas, D., et al., *Three dimensional Monte Carlo code for photon migration through complex heterogeneous media including the adult human head.* Optics Express, 2002. **10**(3): p. 159-170.
46. Tong, Y. and B.d. Frederick, *Time lag dependent multimodal processing of concurrent fMRI and near-infrared spectroscopy (NIRS) data suggests a global circulatory origin for low-frequency oscillation signals in human brain.* NeuroImage, 2010. **53**(2): p. 553-564.
47. Tuchin, V., *Tissue Optics: Light Scattering Methods and Instruments for Medical Diagnosis*. 2007.
48. Vermeij, A., et al., *Very-low-frequency oscillations of cerebral hemodynamics and blood pressure are affected by aging and cognitive load.* NeuroImage, 2014. **85, Part 1**(0): p. 608-615.
49. Flanders, M., *What is the biological basis of sensorimotor integration?* Biological Cybernetics, 2011. **104**(1-2): p. 1-8.

Biographical Information

Pallavi Natekar was born in Mumbai, India on April 1st, 1987. She completed her Bachelors in Instrumentation and Control Engineering from the University of Pune, India in May 2009. She started her career as a Business Operations Associate at ZS Associates and worked primarily on analyzing data for a North American pharmaceutical giant. She then worked with Veol Labs on developing innovative medical solutions. During this period, she decided to expand her horizons in healthcare and moved on to pursue Masters in Biomedical Engineering from the University of Texas at Arlington. Her research work during the graduate course involved image and signal processing, Monte Carlo simulations for optical diffusion. She further plans to pursue her interests by taking up challenging full time engagements in healthcare and allied fields.